Universidad Carlos III de Madrid

Departamento de Tecnología Electrónica

TRABAJO FIN DE GRADO

SISTEMA DE RECONOCIMIENTO BIOMÉTRICO MEDIANTE FIRMA MANUSCRITA ON-LINE 3D

Autor: Marlon Alberto Granda Carrillo

Titulación: Grado en Ingeniería Electrónica Industrial y Automática.

Tutor: Luis Mengibar Pozo

Madrid, 15 de junio de 2013

Agradecimientos

En este momento quiero expresar mi profundo agradecimiento a todas las personas que han contribuido en mi desarrollo tanto personal como intelectual. No puedo nombrar a todos, pero si quiero reconocer el valor de algunos de ellos.

A mi familia, por el apoyo y confianza depositada en mí, han sido un pilar en el que apoyarme, respaldándome para alcanzar mis objetivos.

A mi tutor D. Luis Mengibar Pozo, por su disposición a ayudarme en todo momento y a todas sus aportaciones durante el desarrollo de este proyecto.

A todos mis amigos, los que hecho a lo largo del grado y los de toda la vida, gracias por estar siempre ahí y hacer que esta etapa sea un recuerdo maravilloso.

Resumen

Este trabajo fin de grado trata sobre el desarrollo de un sistema de reconocimiento biométrico. Como patrón biométrico se utiliza una secuencia de movimientos realizados con la mano que denominaremos firma 3D por su analogía con la firma manuscrita o 2D. Como algoritmo de reconocimiento se utiliza DTW (Dynamic Time Warping).

Para la realización de la firma se hace uso del microcontrolador STM32F303VCT6 de núcleo ARM Cortex M4. Los núcleos de ARM son ampliamente utilizados a nivel mundial, al encontrarse disponible actualmente en casi cualquier dispositivo móvil como tabletas o teléfonos móviles e incluso ordenadores de potencia media.

Junto al microcontrolador se utilizan dos sensores con los que se obtienen los datos necesarios para realizar el reconocimiento biométrico, siendo la función principal del microcontrolador la lectura y envío de los datos extraídos durante la realización de la firma manuscrita, siendo el receptor de estos valores el ordenador.

Se realizan pruebas de recepción de datos con el programa hyperterminal para asegurar la comunicación y el correcto funcionamiento del microcontrolador a través del puerto USB. Para evitar el tener que utilizar programas de terceros se ha desarrollado un programa para la recepción y almacenamiento de datos que reemplaza al hyperterminal. Este programa se ha desarrollado usando el lenguaje de programación Visual Basic y empleando el entorno de desarrollo Visual Studio 2012, que al ser de la empresa Microsoft asegura el poder ejecutarse en cualquier ordenador personal que disponga del sistema operativo Windows. Además este programa establece la comunicación con el microcontrolador para la recepción de datos, permitiendo realizar la verificación de la firma o el almacenamiento de la misma en la base de datos asociándole el nombre del usuario que la ha realizado.

Un segundo programa desarrollado es el de aplicación del algoritmo DTW y decisión, cuya función es la de leer los valores de la firma del usuario de la que se desea verificar desde la base de datos y compararla con la capturada, realizando dicha comparación mediante la aplicación del algoritmo DTW, el cual nos da un resultado el que posteriormente permite decidir al sistema la aceptación o rechazo de la firma. Este programa está desarrollado en el lenguaje de programación C# utilizando el entorno de desarrollo Visual Studio 2012.

Se analizan los resultados obtenidos, verificando la relevancia de las características que tenemos disponibles, lo que evita introducir parámetros que puedan alterar el sistema haciéndolo menos fiable o que aporten poco en el proceso de reconocimiento. También se determina el umbral de decisión del sistema mediante los parámetros FAR y FRR, con el objetivo de disminuir los posibles falsos rechazos y las falsas aceptaciones. Además, se analiza el comportamiento del sistema con un menor número de características, comprobando si se mantiene igual de fiable o la misma se ve deteriorada debido a esta acción. Por último, se comprueba que el sistema de reconocimiento desarrollado funciona de forma correcta, reconociendo a los usuarios que forman la base de datos y rechazando a los posibles impostores.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo I

Introducción y Objetivos

1.1 Introducción

Actualmente un proceso de reconocimiento de personas se puede realizar usando parámetros biométricos como la huella dactilar, el rostro o el iris, para diversas aplicaciones como el acceso a servicios telemáticos, cajeros, compra por Internet, acceso a lugares restringidos, etc. De los cuales, algunos como el reconocimiento por iris se encuentra muy limitado a las grandes industrias, por lo que no es algo común que se aprecie en cualquier lugar, pero según avanza la tecnología también evoluciona la forma en la que se obtiene, por ejemplo, acceso a lugares determinados y en las que solo unas cuantas personas pueden acceder.

Con el tiempo se ha visto como uno de las formas más usadas en el mundo para el reconocimiento de personas, la firma manuscrita, ha ido evolucionando, se ha visto como actualmente conviven el firmar de la forma convencional a realizar la firma, ya no sobre el papel, sino en tabletas digitalizadoras que obtienen diversos parámetros que luego son procesados y comparados con los almacenados en una base de datos para su verificación y determinar si la persona es la que dice ser o identificarlo recorriendo la base de datos.

La firma 2D, permite la autenticación de personas siendo está actualmente ampliamente aceptada para la acreditación personal en la vida cotidiana, no siendo un sistema de verificación exento de falsos positivos [3]. Dando un paso más hacia el desarrollo de la firma como medio de autentificación, se está reemplazando las tabletas digitalizadoras por dispositivos que cuentan con otros tipos de sensores, como el giroscopio y el acelerómetro, que actualmente lo incluyen casi en su totalidad cualquier dispositivo móvil como tabletas o teléfonos móviles.

El uso de este tipo de dispositivos está muy expandido en el mundo, además de que son más pequeños que una tableta digitalizadora, en el caso del teléfono móvil, en donde se reemplazan los parámetros que se obtienen en 2D por parámetros en 3D, como la profundidad o el ángulo de giro en un eje determinado dados por los sensores que incluyen [4]. Se sustituye la realización de la firma sobre un soporte físico, a realizarla mediante un gesto en el aire, por lo que pasaríamos a obtener una dimensión más siendo esta la firma 3D.

Mediante la firma 3D, obtenemos más características del comportamiento del individuo, de los cuales se hablará durante el desarrollo de la memoria del proyecto. Al igual que la firma 2D, se contará con una base de datos, en la que los valores obtenidos serán procesados para su posterior comparación y determinar la autenticidad de la persona que realiza la firma. Así mismo este proceso no está exento de falsos positivos, pero estos se ven reducidos al contar con un mayor número de características extraídas de la firma 3D para su posterior verificación.

Durante el desarrollo de este trabajo, se utiliza una tarjeta con los sensores necesarios para obtener los parámetros de la firma 3D, llegando a ser más rápido y fiable, al contar con una capa menos como puede ser el sistema operativo que integran los teléfonos móviles y al ser utilizado específicamente para un cometido como lo es la obtención de los datos a través de los sensores, evitando así que se ejecuten procesos secundarios y que vuelvan más lento al proceso de captura de datos.

1.2 Objetivos

Durante el proceso de desarrollo del sistema de reconocimiento biométrico mediante firma 3D, se han establecido varios objetivos a alcanzar, siendo cada uno de ellos los pasos a dar hasta la consecución final del mismo.

- **Comunicación de la tarjeta del microcontrolador STM32F303VCT6 con el ordenador.**

La tarjeta a utilizar dispone de un microcontrolador STM32F303VCT6 de núcleo Cortex M4 de la empresa ARM, dicha tarjeta cuenta con dos puertos mini USB, siendo uno para la programación y depuración de los programas y el segundo para la utilización como puerto USB de comunicación de libre uso. Como primer paso, está el conseguir la comunicación entre la tarjeta de desarrollo y el ordenador, por el cual se enviarán los datos obtenidos de los sensores incluidos en la placa de desarrollo. Para verificar su correcto funcionamiento se ha utilizado el programa HyperTerminal siendo este luego reemplazado por uno de desarrollo propio.

- **Programación del microcontrolador para la obtención de datos.**

Para la programación del microcontrolador se utilizará el entorno de desarrollo Atollic TrueStudio, así como las librerías disponibles en el mismo. En este paso se realizará la programación del microcontrolador, configurando los sensores disponibles, el giroscopio y el acelerómetro, así como también se establecerá el tiempo de adquisición de los datos, para posteriormente ser procesados y enviados al ordenador por vía USB.

- **Desarrollo de un programa en Visual Studio mediante el cual se almacenan los datos capturados.**

Utilizando Visual Studio 2012, se desarrollará un programa de consola en lenguaje Visual Basic, el cual obtendrá los datos enviados vía USB por el microcontrolador, dando la opción de ser almacenados en la base de datos o guardados para su posterior verificación con los valores almacenados en la base de datos.

- **Desarrollo de un programa en Visual Studio para la implementación del algoritmo DTW y verificación de firma 3D con la base de datos.**

Utilizando Visual Studio 2012, se desarrollará una aplicación de Windows en lenguaje C#, el cuál aplicará el algoritmo DTW (Dynamic Time Warping) a los datos disponibles en la base de datos y a los valores que se han obtenido con el programa anterior, los resultados obtenidos se analizarán para su verificación y determinar si ha sido posible verificar al usuario en la base de datos mediante la firma.

- **Verificar el poder discriminante de las características, establecer la tasa de error y la posibilidad de utilizar un menor número de características en el sistema de reconocimiento.**

Utilizando los datos obtenidos, realizar un análisis del poder de discriminación de las características. Posteriormente obtener los resultados de aplicar el algoritmo DTW para establecer la tasa de error del sistema de reconocimiento. Por último analizar como varía la precisión del sistema al disminuir el número de características y si esto es posible.

Capítulo 2

Estado de la técnica

2.1 Reconocimiento biométrico

En el reconocimiento biométrico se pueden utilizar dos tipos de patrones o señales biométricas para la identificación de personas: la biometría estática y la biometría dinámica.

La biometría estática se refiere a características que no varían con el tiempo o que su evolución es muy lenta, tales como el iris ocular o la huella dactilar. Mientras que la biometría dinámica se refiere a características basadas en el comportamiento de las personas, como es el caso de la firma manuscrita y la voz [1].

2.1.1 Reconocimiento por Huella dactilar

Las huellas dactilares son únicas en cada persona, incluidos los gemelos idénticos tienen huellas dactilares distintas que los identifican entre ellos. El reconocimiento biométrico por huella dactilar se basa en el análisis de la marca que produce el dedo sobre una superficie lisa, la impresión de esta marca se debe a que el dedo presenta en su anverso una serie de crestas, conocidas como crestas papilares, que conforman un dibujo muy característico [1].

Al ser un parámetro fisiológico el nivel de fiabilidad para el reconocimiento es elevado, esto debido a que los dibujos que forman la impresión de la huella dactilar son permanentes ya que permanecen invariables en número, situación forma y dirección. Inmutables, debido a que las crestas papilares no pueden modificarse fisiológicamente y diversiformes, es decir, que presentan diversidad de formas.

En cada huella existen regiones en donde son apreciables cambios en las crestas, cuando la misma se divide en dos o convergen en una, donde empieza y donde termina, todos estos puntos de interés son llamados comúnmente minucia, siendo estas características las que se extraen para posteriormente ser utilizadas en el reconocimiento de las personas.

De una minucia se pueden obtener dos características: la posición en coordenadas x,y; y la orientación. El sistema utilizado para el reconocimiento biométrico mediante huella dactilar, al igual que en la firma manuscrita y en general que en todos los sistemas de reconocimiento biométrico, necesitan de un algoritmo que extraiga y compare los datos de las crestas papilares con los presentes en una base de datos, este algoritmo realiza un filtrado, el mismo que elimina la información no necesaria para extraer solo las minucias, es decir, los puntos de interés y posteriormente compara dichos valores con los conocidos para su reconocimiento.

2.1.2 Reconocimiento por Rostro

Para el reconocimiento de personas mediante el rostro se realiza un análisis de la estructura general, la forma y las proporciones de la cara. Las características extraídas en el reconocimiento de rostro son usualmente las distancias entre los distintos órganos como ojos, nariz y boca, el área de cada uno de ellos, el ángulo formado entre órganos, entre otros [1].

Uno de los inconvenientes que presenta este sistema de reconocimiento es que depende de la iluminación, el ángulo, la expresión y la edad de la persona, por lo que para conformar una base de datos sólida, es necesario la obtención de distintas imágenes, con expresiones y ángulos diversos que permitan tener un sistema útil en el reconocimiento. Además sería necesaria la actualización periódica de la base de datos, esto debido a la dependencia de la efectividad del reconocimiento de la persona respecto a su edad.

A diferencia de otros parámetros biométricos, el de reconocimiento de rostro sería el más intrusivo, al ser necesario colocar la cara sobre el equipo electrónico que realiza la captura de imagen para su posterior extracción de características y reconocimiento, debido a esto este método no es muy aceptado en entornos que requieran un alto nivel de seguridad.

2.1.3 Reconocimiento por Iris

El iris es un tejido pigmentado de alta movilidad y que es visible desde el exterior, debido a la transparencia de la córnea que además cumple la función de proteger al órgano de agentes externos.

Al igual que en la huella dactilar, el iris se encuentra bien diferenciado en hermanos gemelos debido a que poseen patrones distintos, así como también los dos ojos de una misma persona presentan patrones diferentes, algo a tener muy en cuenta a la hora de realizar la adquisición de datos para el almacenamiento en la base de datos y su posterior reconocimiento, ya que esta diferenciación haría que para el reconocimiento fuese necesario el tener que usar siempre el mismo ojo, empleado para adquirir los valores que se han almacenado en la base de datos o para asegurar su correcto funcionamiento tener los datos de ambos ojos [8].

Al ser un método extremadamente diferenciador, se podría asegurar unas tasas de falsa aceptación prácticamente nulas, lo que garantiza su viabilidad para ser utilizada en entornos de autentificación de alta seguridad.

La identificación biométrica por iris consta de cuatro etapas, detalladas a continuación [8]:

1. Captura de los datos fisiológicos.
2. Pre-procesamiento de los datos capturados.
3. Extracción de características propias de cada usuario.
4. Verificación de las características extraídas con los datos previamente almacenados en la base de datos.

En el pre-procesado de la extracción de características para el reconocimiento tenemos: la localización del iris dentro de la imagen, la detección de los bordes del iris, la eliminación de las partes de la imagen no deseadas, compensación del tamaño del iris, debido a la distancia del sujeto respecto al objetivo y la dilatación o contracción de la pupila.

Una de las ventajas que presenta el reconocimiento por iris es su dificultad a la hora de intentar un fraude, esto es debido a que los sistemas empleados para este tipo de reconocimiento capturan dos o más imágenes del ojo humano, centrándose en el iris, al que le han aplicado distintos niveles de luz para comprobar las respuestas a la misma, por lo que una foto o un ojo de plástico con el iris pintado serían rechazados debido a su nula respuesta a los cambios de luz.

Como inconvenientes de este método tenemos que es un sistema intrusivo, que podría incomodar al usuario, el alto coste de los equipos necesarios para su funcionamiento así como también la necesidad de un ordenador de altas prestaciones ya que requiere de un elevado coste computacional.

2.1.4 Reconocimiento por Voz

El reconocimiento por voz es también uno de los métodos utilizados para la autentificación de personas que junto a la firma manuscrita está dentro del reconocimiento biométrico por comportamiento de las personas. Se ha podido establecer que los patrones y las frecuencias con los que cada persona dice una misma palabra son diferentes, haciéndolos únicos.

Cada palabra se descompone en segmentos, los cuales tienen tres o cuatro tonos dominantes que son capturados en forma digital y que se emplean en una tabla o espectro, para conformar el llamado voiceprint, que en este caso son las características extraídas de la voz y almacenadas en una base datos en donde cada frecuencia dominante se expresa como un dato binario.

De esta forma, para el reconocimiento de una persona, la misma pronuncia su frase de acceso, siendo los fragmentos comparados con los disponibles en la base de datos y dependiendo del valor de coincidencia se decide su aceptación o rechazo, siempre antes aplicando un algoritmo, ya que como se ha expuesto antes, nunca se tendrán dos muestras exactamente iguales.

Presenta la ventaja de ser un método poco intrusivo, al ser solo necesario decir una frase o palabra, sin intervenir en el entorno del usuario.

Como inconvenientes el reconocimiento por voz presenta una alta sensibilidad al ruido ambiente además de las variaciones físicas de la persona, por ejemplo, si está constipado, lo que alteraría los patrones extraídos, obteniendo como resultado un falso rechazo por parte del sistema de reconocimiento. Por último, no es tan seguro como otros sistemas biométricos por lo que se lo suele utilizar en conjunto con otros sistemas como puede ser el reconocimiento por rostro o iris.

2.1.5 Reconocimiento por Firma manuscrita

La firma manuscrita dentro del reconocimiento biométrico, es la más utilizada en el mundo como medio de autentificación, teniendo aceptación legal [1]. Según avanza la tecnología se ha visto el desarrollo de la misma, de pasar de firmar sobre el papel a realizar la firma sobre una tableta digitalizadora y ahora el desarrollo de la firma 3D mediante gestos, con un dispositivo que cuente con los sensores necesarios para la adquisición de los datos, que definen el comportamiento de la persona y que posibiliten su reconocimiento.

En la actualidad existen dos métodos para la verificación de la firma: off-line o estáticos y on-line o dinámicos. El método off-line se basa en escanear una firma hecha sobre el papel para posteriormente obtener determinados puntos que caractericen a esa firma, la verificación de la firma mediante el método off-line es considerado más complicado que en la firma on-line [2]. En cambio el método on-line se basa en extraer las características dinámicas de la firma, dependiendo de si esta es 2D o 3D.

Si comparamos los métodos off-line con los on-line, se puede apreciar un mayor número de características disponibles en el segundo, ya que el método off-line sólo nos permitirá obtener información de la posición de la firma sobre unos ejes determinados al obtener la imagen de la firma escaneada de un papel, en cambio en el método on-line, dependiendo de los sensores de los que disponga el dispositivo empleado para el desarrollo de la firma, se puede obtener un

mayor número de características como la velocidad, aceleración o velocidad angular, lo que los hacen más fiables que los sistemas off-line.

En los sistemas on-line, los sensores proveen, además de determinados parámetros de la firma, información sobre el acto propio de firmar que se relaciona con el usuario específico, siendo esta una característica de comportamiento y debido a este factor los sistemas on-line se consideran más fiables frente al fraude [3].

En la firma 2D, se dispone de información como la presión ejercida, la inclinación, posición y velocidad del stylus durante la realización de la firma, mediante el empleo de una tableta digitalizadora [3]. En la firma 3D, la presión ejercida es remplazada por la profundidad, es decir, mientras que en la firma 2D solo se tienen dos ejes de posición x e y, en la firma 3D pasamos a tener una dimensión más, ya que el movimiento de la mano que da lugar a la forma 3D se desarrolla en el espacio tridimensional.

También tenemos que dependiendo del tipo de sensores disponibles en el dispositivo para la realización de la firma, además de la posición, se puede obtener información de la aceleración o de la velocidad angular respecto a cada eje durante la realización de la firma. Además tenemos que el empleo de la firma 3D gana seguridad frente a la firma 2D, debido a que aporta más información que lo convierte en un sistema más robusto frente al fraude [4].

Debido a que una misma persona no es capaz de reproducir exactamente igual su firma, los datos adquiridos necesitan ser procesados antes de realizar la verificación con la base de datos, existiendo varios algoritmos que realizan esta función, entre los más conocidos tenemos: Modelos Ocultos de Markov (HMM), lógica difusa, redes neuronales o Mezcla de Gaussianas (GMMs)[3],[5], para el desarrollo de este proyecto se utilizará el algoritmo Dynamic Time Warping (DTW) del cual se habla con mayor profundidad en el apartado *2.3 Dynamic Time Warping*.

En la figura 1 se pueden observar las tres principales fases para el reconocimiento biométrico, el mismo no es sólo aplicable a la firma manuscrita, sino también a otros como la huella dactilar o el rostro.

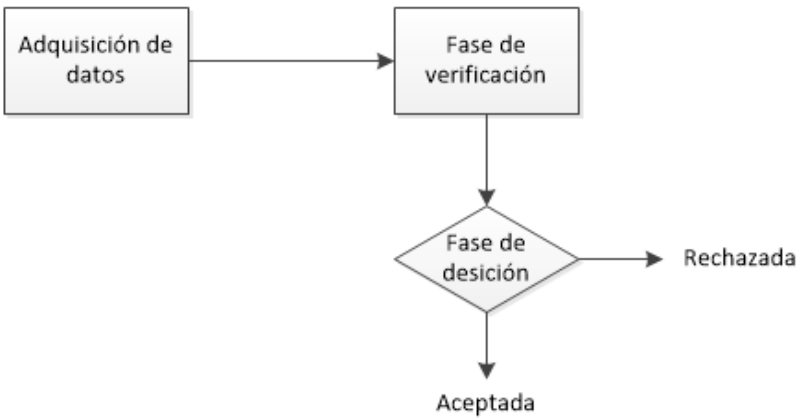

Figura 1. Diagrama de bloques simple de un proceso de reconocimiento

La fase de adquisición de datos engloba los procesos de adquisición de información de la firma así como la identificación de las diversas características, clasificándolas para posteriormente almacenarlas o pasar al proceso de verificación. En la fase de verificación, la información de la fase anterior es comparada con una base de datos mediante el uso de un algoritmo teniendo como resultado un valor, este valor servirá para que en la fase de decisión se determine si la firma es aceptada o rechazada.

Se debe diferenciar entre identificación y verificación de una firma, esto debido a que son procesos distintos. En el proceso de identificación, la firma capturada es comparada con toda la base de datos hasta encontrar el usuario al que pertenece dicha firma, mientras que en el proceso de verificación, la firma capturada es comparada con unos valores específicos de la base de datos para la comprobación de que esa firma pertenece a la persona que la ha realizado, es decir, mientras que el primero es un proceso (1,n) el segundo es un proceso (1,1).

2.1.5.1 Reconocimiento de gestos

La realización de una firma 3D se basa en el desarrollo de gestos en un espacio tridimensional, en donde los sensores con los que cuente el dispositivo de captura permiten obtener las características de la firma. Por lo que una firma 3D se puede explicar como una mezcla de la firma 2D y gestos, donde toma ciertos fundamentos de cada uno de ellos.

En relación a los gestos, Watanabe y Yachida propusieron un nuevo método de reconocimiento de gestos en tiempo real para sistemas interactivos orientados a humanos. Este método obtiene una aproximación de la estructura 3D, que se construye por la expansión KL (Teorema de Karhunen-Loève) de una secuencias de imágenes. Este método, además también obtiene magnitudes como la velocidad y la aceleración, lo que le permite incluso reconocer gestos complicados [20].

En la firma 3D no se ha uso de un secuencia de imágenes pero si de dos características que también se utilizan en el reconocimiento de gestos, es decir, la aceleración lineal y la velocidad angular en nuestro caso, y es aquí en donde se determina que una firma 3D comparte similitudes con un sistema de reconocimiento de gestos.

2.1.5.2 Adquisición de datos

Dependiendo del método utilizado, off-line u on-line, el proceso de adquisición de las firmas será distinto. Para el método off-line las características se extraen de una imagen obtenida de la firma escaneada previamente, de la cual se obtienen determinados puntos representativos que definan a la misma, establecidos sobre un eje de coordenadas.

En el método online las características se obtienen durante la realización de la firma, las mismas dependerán de si la firma es 2D o 3D y del dispositivo empleado, como se ha expuesto en el apartado anterior.

El número de datos disponibles dependerá de la frecuencia con la que se obtengan, estableciendo para ello un periodo de adquisición que nos permita reducir las probabilidades de falsos rechazos y falsas aceptaciones [4], consiguiendo un sistema más sólido frente a fraudes. Otro parámetro que también se puede establecer como característica, es el tiempo empleado en la realización de la firma, ya que conociendo el período de adquisición de los datos y el número de datos, con una simple operación obtenemos el tiempo total empleado,

esta característica puede llegar a ser relevante en el proceso de verificación o simplemente no aportar un mayor nivel de fiabilidad, esto dependiendo del número de características con las que contemos en el proceso.

2.1.5.3 Parámetros FAR y FRR

Como se ha indicado anteriormente la firma manuscrita no está exenta de fraude, por lo que para establecer un umbral que determine la veracidad de la misma se estudian los resultados de las tasas de falsas aceptaciones o FAR y las tasas de falsos rechazos o FRR.

El parámetro FAR indica la proporción de firmas falsificadas reconocidas como firmas genuinas y el parámetro FRR indica la proporción de firmas genuinas clasificadas como firmas falsas, de ahí que un alto número de falsas aceptaciones conlleve a que el sistema sea poco fiable o que un alto número de falsos rechazos harían que el sistema no sea efectivo en la identificación, por lo que se debe encontrar un equilibrio entre ambas, por lo que generalmente se toma la intersección entre las dos curvas conocido como Equal Error Rate o tasa de equierror (ERR), siendo este el punto en el que se igualan los parámetros FAR y FRR, además entre más pequeño sea este valor más preciso será el sistema [6][7]. En la figura 2 se observan las curvas de las falsas aceptaciones y los falsos rechazos así como el ERR.

Las curvas se obtienen variando el umbral, por lo que por cada valor del umbral se obtienen las tasas de falsas aceptaciones y falsos rechazos, siendo el resultado un punto que compone la curva.

En este caso se observa como a valores altos del umbral el parámetro FAR es muy elevado y el parámetro FRR es cero, mientras que para valores del umbral pequeños sucede el caso contrario, es decir, el parámetro FRR toma valores muy altos y el parámetro FAR es cero.

También se observa como el punto en el que se obtiene un equilibrio entre ambos parámetros, FAR y FRR, es el punto de corte entre ambas curvas, de ahí que se tome el valor del umbral asociado al ERR como umbral del sistema para la verificación de firmas.

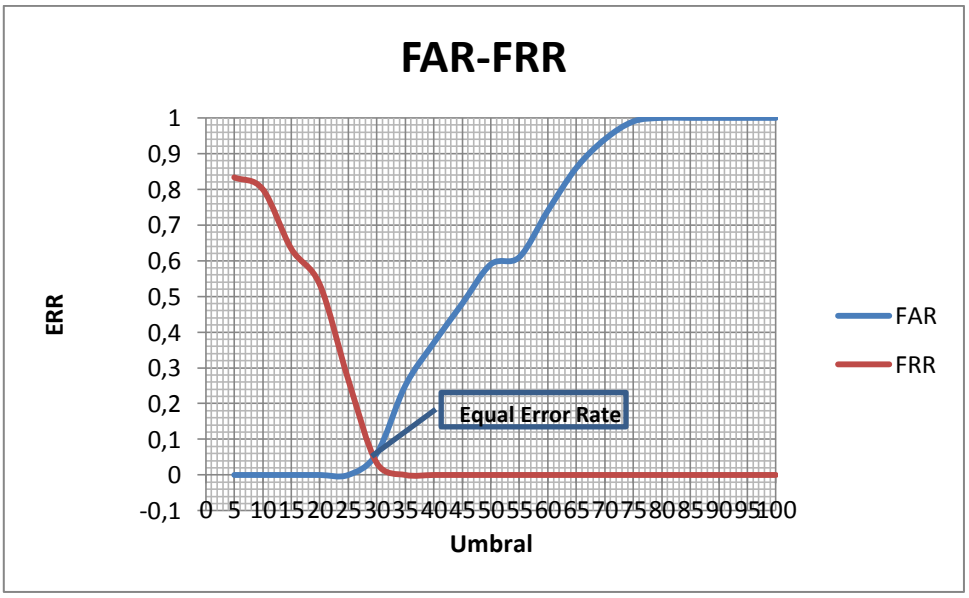

Figura 2. FAR frente a FRR

2.2 Algoritmos de reconocimiento

Actualmente existen varios algoritmos que se emplean para el reconocimiento biométrico, con sus respectivas ventajas e inconvenientes, los más conocidos se detallan a continuación.

2.2.1 Redes Neuronales

La verificación de firmas utilizando las redes neuronales en la actualidad está ampliamente expandida. Sólo con funciones de álgebra lineal simple, un desarrollador puede crear un sistema de redes neuronales artificial sin experimentar problema alguno. Las redes neuronales tienen limitaciones, debido a que los datos o parámetros utilizados deben ser establecidos al principio, si aparecen datos adicionales, una vez que el sistema se encuentra en marcha, es imposible agregarlos, teniendo que empezar de cero para su consideración [9].

2.2.2 Modelo Oculto de Markov

Al igual que las redes neuronales, el Modelo Oculto de Markov (HMM por sus siglas en inglés), también son ampliamente utilizados en la verificación de firmas. Los estudios que analizan la implementación del modelo oculto de Markov para la verificación de firmas son desarrollados mediante la utilización de características globales de la firma. Cada firma se modela utilizando HMM para posteriormente ser utilizados en el proceso de verificación. Este algoritmo también dispone de limitaciones, siendo el más relevante el alto coste computacional del que pueda requerir [10], [9].

2.2.3 Support Vector Machines

Las Support Vector Machines o SVM por sus siglas en inglés, son algoritmos de aprendizaje automático, que utilizan características dimensionadas en el espacio. Las SVM se utilizan para clasificar las características extraídas de la firma que se utilizarán en el proceso de verificación. Las SVM también poseen algunos inconvenientes, como la determinación del kernel a utilizar y que la verificación utilizando las SVM requiere mucho tiempo y espacio [9].

2.3 Dynamic Time Warping

El Dynamic Time Warping o Alineamiento Temporal Dinámico en español ,también conocido como trayectoria de coincidencia técnica, originalmente diseñado por Kruskal y Liberman, se utiliza con éxito en los campos del reconocimiento de voz, reconocimiento de patrones visuales, la robótica, el procesamiento del habla, la fabricación, la medicina y el reconocimiento de firma manuscrita [11].

Puede ser empleado junto con otros algoritmos existentes, en este caso el proceso pasaría a ser un sistema de clasificación múltiple. Pero también se puede utilizar en un sistema de reconocimiento independiente, pasando en este caso a denominarse clasificadores simples [12], siendo este último en el que se basará el proyecto.

Es un algoritmo para medir la desviación entre dos secuencias que pueden variar en el tiempo o la velocidad, por lo que cuanta más pequeña sea la desviación se podría afirmar si esas series temporales se tratan de una misma señal, como puede ser la firma de un individuo, o en caso contrario se tratan de dos señales distintas.

En la figura 3 se muestra de forma genérica la representación de la función de alineamiento.

De donde:

- Serie temporal A, situada en el eje X, es la firma capturada.
- Serie temporal B, situada en el eje Y, es la serie temporal tomada como patrón.
- El vector C_K, es el resultado de la diferencia entre ambas series temporales.

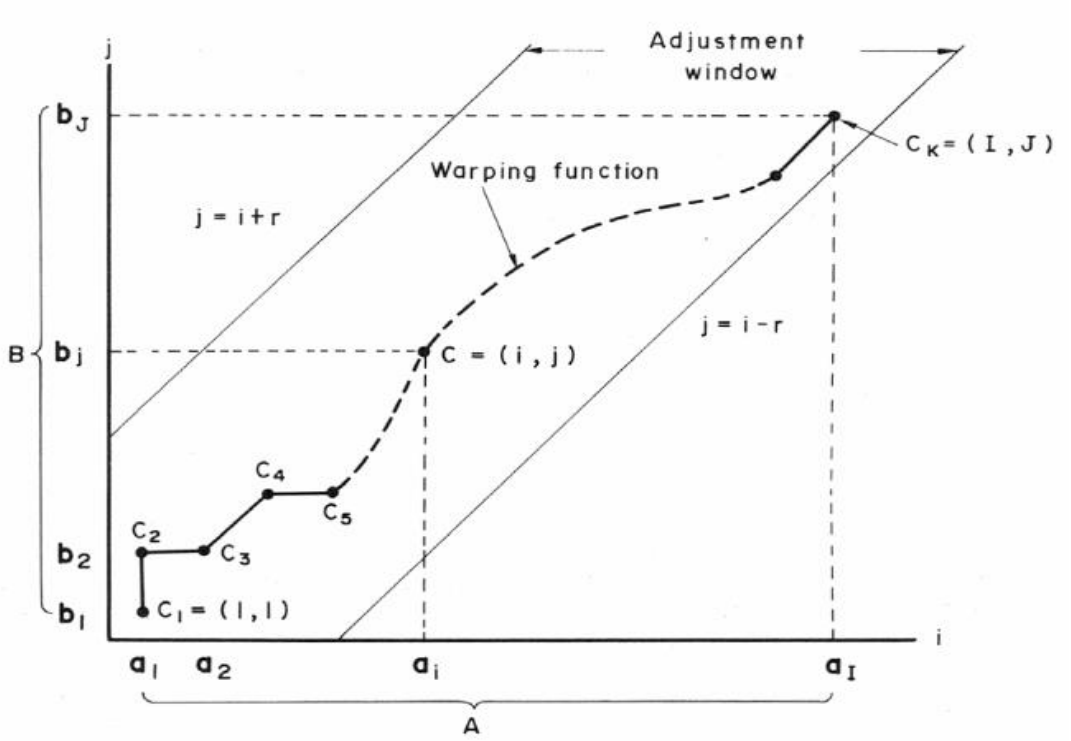

Figura 3. Función de alineamiento entre dos firmas.

De la figura anterior también se puede obtener el coste de la alineación, para ello primero se considera una matriz de dimensión ixj, determinada por el número de datos de los que dispone cada serie temporal, conteniendo cada celda el valor DTW resultado de la comparación de cada punto. Posteriormente nos encontramos con un camino en la matriz que inicia en el punto (1,1) y finaliza en el punto (i,j) de modo que el coste acumulado promedio a lo largo de la ruta se reduce al mínimo. Si la curva pasa por las celdas (i,j), entonces dichas celdas contribuyen al coste(a_i,b_j) para el coste acumulativo [13]. La función de coste puede definirse de modo flexible dependiendo de la aplicación, en este caso utilizamos la ecuación [1].

$$coste(a_i, b_j) = |a_i - b_i|^2 \qquad [1]$$

La curva puede variar varias celdas horizontalmente, a lo largo de *i* o verticalmente a lo largo de *j*, que hace que la coincidencia entre las dos secuencias no sea estrictamente la misma, pero si parecidas. Siendo esta la robustez que proporciona el Dynamic Time Warping para alinear secuencias [13].

Capítulo 3

Entorno de desarrollo

En este capítulo se detallarán las herramientas utilizadas para el desarrollo del sistema de reconocimiento.

Para la programación y depuración del código a ejecutar por el microcontrolador emplearemos el entorno de desarrollo Atollic TrueStudio, mientras que para el desarrollo de las aplicaciones de recepción y almacenamiento de datos y la implementación del algoritmo DTW, se utiliza el entorno de desarrollo Visual Studio 2012.

Además también se expondrá los lenguajes de programación utilizados, según la aplicación, justificando su elección.

En cuanto a la tarjeta de desarrollo se detallan sus componentes principales, como el microcontrolador y los sensores que incluye y que serán utilizados en el sistema de reconocimiento.

3.1 Atollic TrueStudio

El entorno de desarrollo utilizado es Atollic TrueStudio, un software basado en el programa Eclipse de software libre y desarrollado por la Fundación Eclipse, siendo también utilizado por Google para el desarrollo de aplicaciones de su sistema operativo móvil Android. La elección del mismo se ha realizado debido al conocimiento previo de su funcionamiento además de las herramientas que incorpora, como los drivers, que permiten configurar de forma accesible los distintos periféricos que integra el microcontrolador, entre ellos los sensores.

Este programa soporta el desarrollo y depuración de la tarjeta del microcontrolador que se utilizará en el proyecto, la STM32F3Discovery, del cual se hablará más adelante. Una de las características interesantes y útiles que incluye este software es la de permitir la depuración paso a paso, acceso a los registros del microcontrolador y variables de usuario, lo que ayuda a la detección de errores que se produzcan durante la ejecución del código.

Permite escribir el código en lenguaje C y C++ estando este último solamente disponible para la versión de pago. El uso de este lenguaje de programación junto con los drivers que incluye hace prescindible el tener que acceder directamente a los registros del microcontrolador, por lo que la programación se realiza a un alto nivel. Atollic TrueStudio está centrado en el desarrollo de programas exclusivos para los microcontroladores a los que da soporte, es decir, no es útil para el desarrollo de aplicaciones de ordenadores, por lo que para la producción de los programas que se emplearán en el proyecto, se utiliza Visual Studio 2012.

3.2 Visual Studio 2012

Para el almacenamiento y verificación de las firmas se hace uso de un ordenador al ser necesario disponer de un equipo capaz de ejecutar programas que requieran un mayor nivel de procesamiento.

Para el desarrollo de estos programas se utiliza el entorno de desarrollo Visual Studio 2012 de Microsoft, el mismo que se ejecuta sobre el sistema operativo Windows 8, también propiedad de Microsoft, lo que nos asegura una compatibilidad plena entre software y hardware.

Este programa también permite la depuración de código paso a paso del software que se desarrolle, siendo útil a la hora de buscar errores en el código. Es una herramienta muy

completa y potente, capaz de soportar varios lenguajes de programación, siendo esta una de las razones de su elección. Como luego se verá para el desarrollo de los programas, se ha utilizado dos lenguajes de programación, Visual Basic y C-Sharp, esto debido a la integración que ofrece el primero a la hora de comunicarse con los puertos del equipo y el segundo debido a las librerías que proporciona Microsoft y que ayudan al desarrollo de una aplicación de Windows.

Los proyectos desarrollados en esta plataforma son *programas de comandos*, lo que permite una interacción simple del usuario con el ordenador.

3.3 Lenguajes de programación C-Sharp y Visual Basic

El lenguaje de programación C-Sharp o C#, es un lenguaje de programación orientado a objetos basado en C++ siendo este otro lenguaje de programación. C-Sharp ha sido estandarizado y desarrollado por Microsoft, las diferencias entre ambos es la inclusión en C# de modelos de objetos de la plataforma .NET también propiedad de Microsoft y que facilita a los programadores el desarrollo de programas utilizando los recursos del sistema. En este lenguaje es en el cual se encuentra desarrollada la librería del algoritmo DTW a implementar, por lo que el programa de aplicación del algoritmo se desarrollará bajo este lenguaje de programación.

En cuanto a Visual Basic, es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Visual Basic también hace uso de modelos de objetos del .NET Framework, este lenguaje además de permitir el desarrollo por líneas de código, proporciona un ambiente de desarrollo completamente gráfico que facilita la creación de interfaces gráficas. Permite la creación de pequeños programas hasta software complejo, es decir no tiene limitación. La última versión es Visual Basic 6.0 publicada en el año 1998, del cual el soporte estándar finalizó en el año 2005 y su soporte extendido en el año 2008 por parte de Microsoft.

Se utiliza este lenguaje de programación para el desarrollo del programa de recepción y almacenamiento de datos, debido a las herramientas que proporciona a la hora de establecer comunicación con los puertos del ordenador, además de que también provee facilidades para el desarrollo de aplicaciones de base de datos, necesaria en nuestro proyecto.

La utilización de dos lenguajes de programación no suponen un inconveniente, ya que C# y Visual Basic son lenguajes de fácil transición entre ellos, pero cada uno cuenta con sus respectivas ventajas, lo que han hecho que sean elegidos para el desarrollo de nuestro software.

3.4 El Microcontrolador

Antes de detallar las características técnicas del STM32F303VCT6 utilizado en el proyecto, primero se describen las diferencias entre un microprocesador y un microcontrolador, que aunque a veces se usan estos términos de forma indiferente para referirse a un mismo componente, en realidad no son iguales.

Un **microprocesador** es un chip o circuito integrado, que integra los elementos de una CPU:

- Unidad de Control.
- Unidad de Proceso:
    - Unidad de Aritmética Lógica o ALU por sus siglas en ingles.
    - Ruta de datos (Data Path), siendo este los registros internos conectados por buses.

Un **microcontrolador** es un circuito integrado que, además de llevar un microprocesador incluye:

- Memorias de programa y de datos.
- Periféricos de entrada/salida, E/S.

Por lo que la diferencia principal se encuentra en que un microcontrolador contiene un microprocesador incluyendo además periféricos que complementen el funcionamiento del mismo. Por lo que el STM32F303VCT6 es un microcontrolador completo en cuanto a periféricos y configuraciones se refiere.

3.4.1 Microcontrolador STM32F303VCT6

El microcontrolador STM32F303VCT6 es un circuito integrado de 32-bits basado en el procesador ARM Cortex M4, el mismo incluye una Microcontroller Unit o MCU, que integra un Procesador Digital de Señal (DSP) y una Unidad de Coma Flotante (FPU), esta última como su nombre indica, está especializado en el cálculo de operaciones en coma flotante.

Al contar con un módulo dedicado a este tipo de cálculos, se gana eficiencia a la hora de procesarlos, ya que cuando no se dispone de una FPU, el microcontrolador realiza las operaciones por software, emulando una operación en coma flotante a través de rutinas software que ejecuta la unidad aritmética lógica (ALU), lo que se traduce en una pérdida de la velocidad y que en aplicaciones en donde se requiera realizar operaciones con valores tipo float o double la carga de procesamiento sería elevada, de ahí la ventaja de que este microcontrolador incluya una unidad específica encargada de este tipo de operaciones y que nos permite ganar tiempo, por lo que es mejor la realización de los cálculos por hardware antes que por software.

Por lo que al disponer de una FPU se gana velocidad en los cálculos y precisión, ya que cuando se trabaja sin FPU y se usa coma fija, que sería el caso si utilizáramos la ALU, habría que considerar los errores por redondeo.

En cuanto al procesador digital de señales (DSP) es un conjunto de instrucciones, con un hardware y software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad, por lo que uno de sus usos comunes es el procesado y representación de señales digitales en tiempo real.

Una de las características que hacen interesante el uso del núcleo Cortex M4 es la disponibilidad de un Controlador de Interrupciones Anidadas o NVIC, que asegura una estructura básica de ARM independiente del fabricante, es decir, todos los microcontroladores que utilicen el mismo núcleo compartirán el mismo sistema de interrupciones.

La diferencia entre las interrupciones de ARM y las que el propio fabricante decida incluir, es que la NVIC solo asigna una interrupción global por periférico, mientras que las del fabricante pueden ser varias, dependiendo de cuál queramos utilizar, así por ejemplo, en un timer se puede habilitar interrupciones por actualización del registro, interrupción por input capture o interrupción por output compare.

Se debe tener presente que en la configuración de las interrupciones de los periféricos siempre es necesario habilitar por código, tanto las interrupciones globales a través de la NVIC como las interrupciones propias de cada periférico.

Este microcontrolador también dispone de timers de 16 bits con auto-recarga, el mismo tiene utilidades para varios propósitos como: medir intervalos de tiempo contando los períodos de reloj transcurridos, medir la duración de un pulso de entrada externo o generar una onda de salida. Se utiliza en el proyecto para la captura de valores de los distintos sensores, es decir, define un intervalo de tiempo entre tomas de datos.

Un periférico necesario para la comunicación entre microcontrolador y ordenador es la USART. La USART o Universal Synchronous Asynchronous Receiver Transmitter, es un dispositivo de entrada/salida capaz de mantener comunicaciones serie síncronas y asíncronas. Este microcontrolador permite su implementación en modo SPI o I2S según la aplicación del mismo.

Además cuenta con otros periféricos que aunque no se utilicen en el proyecto es interesante conocer la disponibilidad de los mismos en el microcontrolador. Estos periféricos se detallan a continuación.

Otro periférico con los que cuenta este microcontrolador es el módulo de Acceso Directo a Memoria o DMA, que permite que ciertas unidades del microcontrolador puedan acceder directamente a la memoria del sistema para leer o escribir independientemente de la unidad central de procesamiento (CPU), lo que permite liberar de carga a la CPU y ganar velocidad en el proceso.

Este microcontrolador también dispone de un Conversor Analógico/Digital, lo que permite que un sistema digital pueda interactuar con un entorno analógico. Convirtiendo una magnitud física en un dato digital, lo que posibilita posteriormente ser procesado por el núcleo u otros periféricos del microcontrolador.

También dispone de un módulo de Conversión Digital/Analógico, el funcionamiento de este periférico es el inverso que el del ADC, ya que genera una tensión de salida analógica siendo esta proporcional al valor digital de sus entradas, es utilizado en sistemas de sonido e imagen entre otros.

Al ser un microcontrolador basado en la arquitectura ARM Cortex M4, siendo este el núcleo del circuito integrado, necesita de una serie de periféricos que aumenten las funcionalidades que el mismo puede ofrecer. En la figura 4 se detallan todos los periféricos que incluye este microcontrolador.

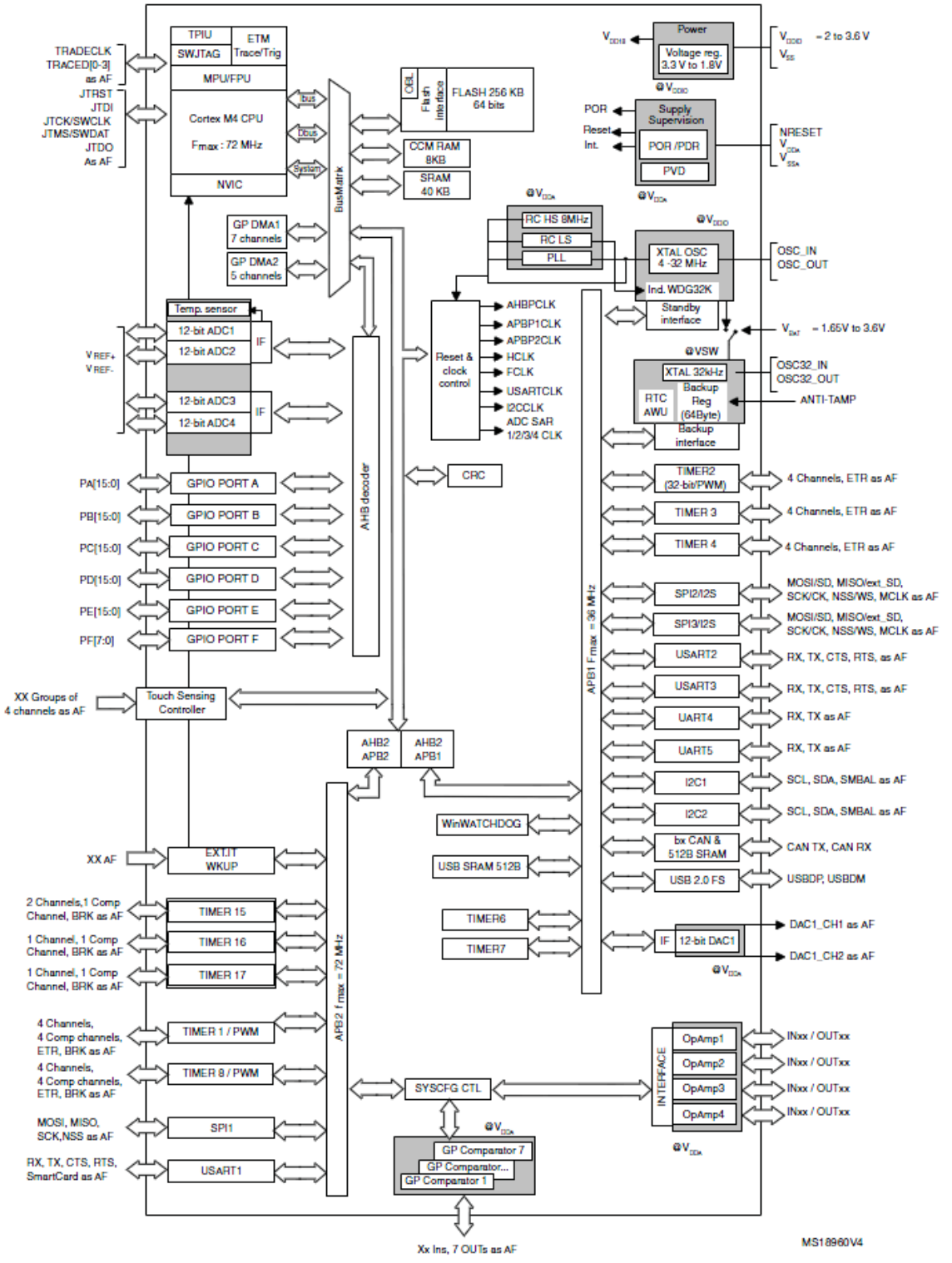

Figura 4. Diagrama de bloques del STM32F303VCT6

3.4.1.1 Características técnicas del STM32F303CVT6

Este microcontrolador cuenta con una gran cantidad de recursos interesantes, entre los más relevantes tenemos:

- Núcleo ARM Cortex M4 de 32 bits con velocidad de hasta 72 MHz con DSP y FPU.
- 256 Kbytes de memoria Flash.
- 40 Kbytes de memoria SRAM.
- Soporta un rango de alimentación de 2.0V a 3.6V.
- Compatible con osciladores de cristal de 4 a 32 MHz con opción de PLL, un oscilador RTC de 32KHz, un RC interno de 8MHz con PLL y un oscilador interno de 40 KHz.
- 12 canales DMA
- 39 canales ADC con resoluciones de 12, 10, 8 o 6 bits, el rango de conversión es de 0 a 3.6V.
- 12 canales DAC.
- 13 timers:
    - Un timer de 32 bits y dos timers de 16 bits con 4 módulos de Input Capture, Output Compare y PWM.
    - Dos Watchdog timers.
    - Un timer del Systick de 24 bits en modo cuenta descendente.
    - Y dos timers de 16 bits para el control del DAC.
- Calendario RTC con alarma, activación periódica en Stop/Standby.
- Interfaces de comunicación:
    - Interfaz CAN.
    - Dos I^2C con velocidades de hasta 1Mbit/s.
    - Hasta cinco USART/UARTs.
    - Hasta tres SPIs, dos con multiplexado a interfaz I^2S, con velocidad programable.
    - Interfaz USB 2.0 de alta velocidad.
- Serial wire debug, JTAG, Cortex M4F ETM.

Por lo que este microcontrolador dispone de un amplio abanico de posibles usos, gracias a todos los periféricos que incluye y que facilita la interacción del mismo con otros dispositivos.

3.4.1.2 Tarjeta de desarrollo STM32F3DISCOVERY

La tarjeta a utilizar para el desarrollo del proyecto es la STM32F3Discovery, la misma lleva integrada un depurador y programador en circuito el ST-LINK/V2, lo que permite realizar la programación del microcontrolador de una forma sencilla, sin necesidad de otro dispositivo. La misma se realiza por medio del puerto USB, previamente instalando los drivers correspondientes en el ordenador con el que se va a trabajar.

Esta tarjeta de desarrollo incluye una serie de componentes externos al microcontrolador, que nos permite explotar las capacidades del mismo. Entre sus características tenemos:

- Microcontrolador STM32F303VCT6.
- ST-LINK/V2 depurador y programador en circuito.
- Alimentación mediante el puerto USB o alimentación externa de 3V y 5V.

- L3GD20, sensor de movimiento ST MEMS, giroscopio digital de 3 ejes.
- LSM303DLHC, acelerómetro lineal 3D digital y sensor magnetómetro 3D digital.
- Diez LEDs
    - LD1 (rojo) indicador de tarjeta alimentada a 3.3V
    - LD2 (rojo/verde) indicador de comunicación mediante puerto USB.
    - Ocho diodos LEDs disponibles para uso del programador.
- Dos pulsadores, uno de usuario y otro de reset.
- Dos puertos USB, uno para la programación y depuración de programa y otro para uso general por parte del usuario.
- Pines que dan acceso a los diferentes puertos disponible en el microcontrolador.

Esta tarjeta es la adecuada al incluir los sensores: giroscopio, acelerómetro y magnetómetro, siendo los dos primeros los que se van a utilizar para la extracción de características de la firma 3D. Al incluir un puerto USB de usuario, nos permite comunicar el ordenador con el microcontrolador sin necesidad de utilizar el entorno de desarrollo Atollic TrueStudio, es decir, mediante el uso de este puerto, previa configuración, la tarjeta STM32F3Discovery sería reconocido como un periférico por parte del ordenador o incluso que la tarjeta pase a ser el dispositivo principal y que el equipo que se conecte a través del miniUSB sea reconocido como un periférico.

En la figura 5 se observa el diagrama de bloques de la tarjeta de desarrollo, en ella se aprecian los distintos sensores, así como también los LEDs y pulsadores disponibles para el usuario.

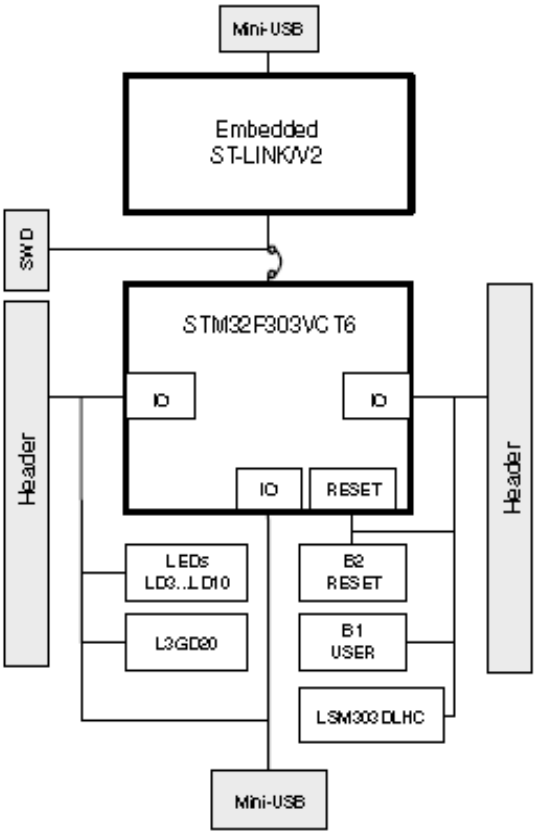

Figura 5. Diagrama de bloques de la tarjeta STM32F3DISCOVERY

3.5 Sensores integrados en la tarjeta de desarrollo

En una firma 2D los datos característicos de la misma se obtienen a través de una tableta digitalizadora (también conocida como tableta gráfica), capaz de muestrear la posición del útil de escritura (stylus) cada cierto tiempo, retornando valores como las coordenadas X,Y, la presión ejercida sobre la tableta o la inclinación del puntero sobre la misma, siendo estos datos posteriormente procesados para ser utilizados en el reconocimiento de la firma manuscrita.

En cambio en la firma 3D, dependiendo del tipo de sensores con los cuente el dispositivo, las características obtenidas pueden variar. El dispositivo del que disponemos, la tarjeta STM32F3Discovery, cuenta con tres sensores: un acelerómetro, un giroscopio y un magnetómetro de los cuales se emplearán los dos primeros. Por lo tanto los datos que podemos obtener de la firma son:

- Las aceleraciones lineales correspondientes a cada eje (x, y, z), medida por el acelerómetro.
- Las velocidades angulares respecto a cada eje (x, y, z), medida por el giroscopio.

Por lo que para la captura de la firma 3D tenemos disponibles 6 características que nos permitirán su reconocimiento. Las características en la firma 3D pasan a ser diferentes a las de la firma 2D ya que en lugar de disponer de las coordenadas X, Y, la presión y la inclinación, se disponen de las aceleraciones y velocidades angulares por cada eje. En la figura 6 se puede observar las ubicaciones de los sensores en la tarjeta de desarrollo.

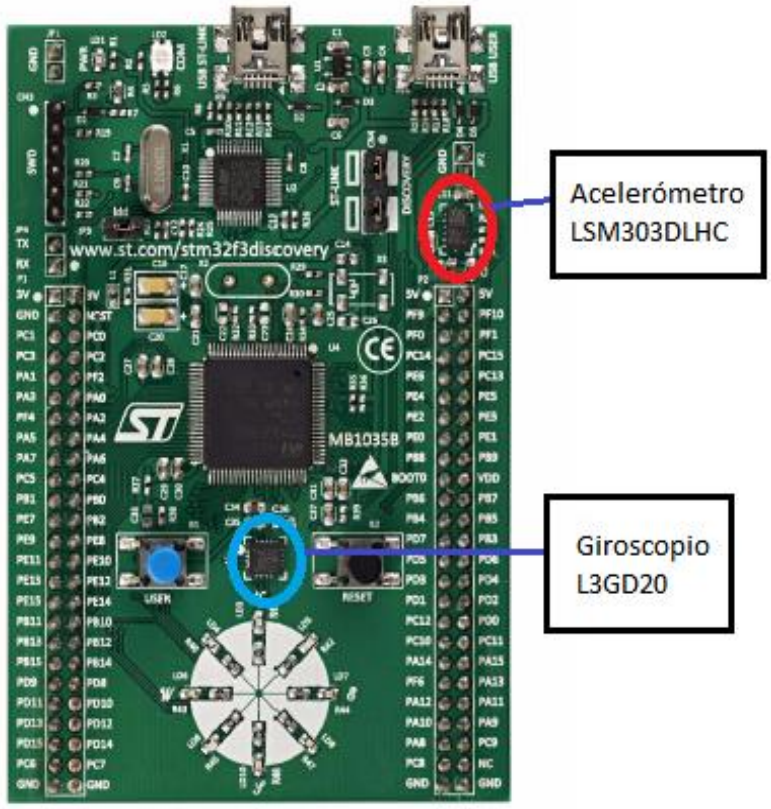

Figura 6. Ubicación de los sensores sobre la tarjeta de desarrollo.

3.5.1 Acelerómetro LSM303DLHC

Los acelerómetros basados en sistemas microelectromecánicos (MEMs), disponen de varias ventajas tales como contar con unas dimensiones reducidas lo que le permite tener un bajo peso, también se caracterizan por la disponibilidad de una alta sensibilidad. Estas ventajas han permitido que estos sistemas se hayan incluido en dispositivos de pequeño tamaño, como en la tarjeta de desarrollo que se utiliza en el proyecto. Cabe destacar que estos dispositivos son ampliamente usados en muchos campos como la orientación inercial o los sistemas de seguridad de los automóviles [15].

Como se ha mencionado anteriormente, la tarjeta STM32F3Discovery incluye un sensor de aceleración en 3 dimensiones, el acelerómetro, permitiendo conocer el sentido y dirección del movimiento que se realiza con el dispositivo.

El LSM303DLHC es un circuito integrado que incluye un sensor digital de aceleración lineal 3D y un sensor digital magnético 3D. Del cual se hará uso del acelerómetro para la obtención de las aceleraciones respecto a cada eje. En la figura 7 se observa el circuito integrado y los ejes de aceleración correspondientes al sensor.

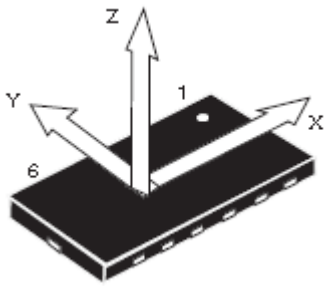

Figura 7. Dirección de aceleraciones detectables.

El sistema incluye una interfaz IC, capaz de medir tanto la aceleración lineal y el campo magnético aplicado sobre el mismo, proporcionando un valor de los mismos a través de un puerto de comunicación I^2C. El entorno de desarrollo Atollic TrueStudio, nos provee de los drivers necesarios para la configuración de este sensor y su posterior obtención de datos.

La sensibilidad de la aceleración lineal configura la ganancia del sensor y puede ser establecida por software. La temperatura tiene poca influencia respecto a la sensibilidad del sensor sobre todo en el entorno en el que se va a utilizar. Cuenta con la ventaja de permitir varias configuraciones de sensibilidad, permitiéndonos establecer la que mejor se adapte a nuestro proyecto.

En la figura 8 tenemos la conexión del circuito integrado que incluye los dos sensores y que se encuentran montados en la tarjeta STM32F3Discovery.

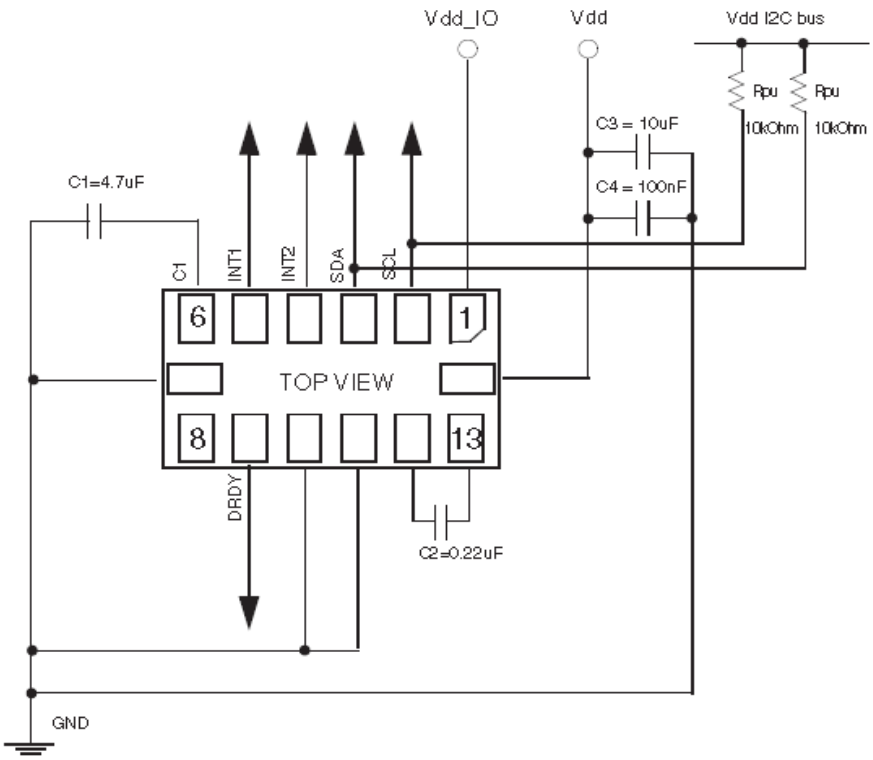

Figura 8. Conexión eléctrica del LSM303DLHC.

Atendiendo a las principales características de este sensor tenemos:

- Detección de campo magnético tridimensional y detección de aceleración tridimensional.
- Sensibilidad del magnetómetro configurable, desde ±1.3 a ±8.1 gauss.
- Sensibilidad del acelerómetro configurable ±2g/±4g/±8g/±16g.
- Puerto de comunicación serie I^2C.
- Tensión de alimentación 2.16V a 3.6V.
- Formato de 16 bits de los datos de salida.

De la tabla 1 se obtiene el rango de temperaturas en los que opera normalmente el sensor, sin que esta tenga una influencia importante en los datos dados por el mismo, siendo este rango de -40°C a 85°C.

Symbol	Parameter	Test condition	Min.	Typ.(1)	Max.	Unit
TSDr	Temperature sensor output change vs. temperature	-		8		LSB/°C(2)
TODR	Temperature refresh rate			ODR(3)		Hz
Top	Operating temperature range		-40		+85	°C

Tabla 1. Rango de temperaturas de operación del LSM303DLHC.

3.5.2 Giroscopio L3GD20

El segundo sensor del que se hará uso para la extracción de características de la firma 3D es el giroscopio, implementado en el circuito integrado L3GD20 y disponible en la tarjeta de desarrollo STM32F3Discovery.

Los giroscopios basados en sistemas microelectromecánicos (MEMS), cuentan con un tamaño reducido y menor peso que otros sensores de velocidad angular tradicionales [16]. La fabricación de este sensor en circuitos integrados, ha permitido el uso del mismo en equipos de pequeño tamaño, posibilitando ampliar sus campos de aplicación, como por ejemplo, en la industria de los teléfonos móviles.

Este sensor en dispositivos móviles es ampliamente usado en videojuegos, con el objetivo de evitar utilizar la pantalla táctil, siendo requerida sólo la rotación del dispositivo para determinar la dirección a seguir por el objeto que controla el usuario, llegando a complementar al acelerómetro, obteniendo como resultado una mejor respuesta y una mayor precisión.

Además el giroscopio nos proporciona información sobre la velocidad angular del dispositivo sobre el que se encuentra montado, produciendo una salida positiva cuando el giro se produce en sentido anti-horario y negativa cuando el giro se produce en sentido horario, siempre respecto al eje considerado para la medición.

En el proyecto se considera la información recogida de la velocidad angular respecto a cada eje, obteniéndose tres características de la firma, es decir, la velocidad angular respecto al eje x, la velocidad angular respecto al eje z y la velocidad angular respecto al eje y. En la figura 9 se aprecia el circuito integrado y los ejes sobre los que se medirá la velocidad angular del dispositivo en el que se encuentra integrado.

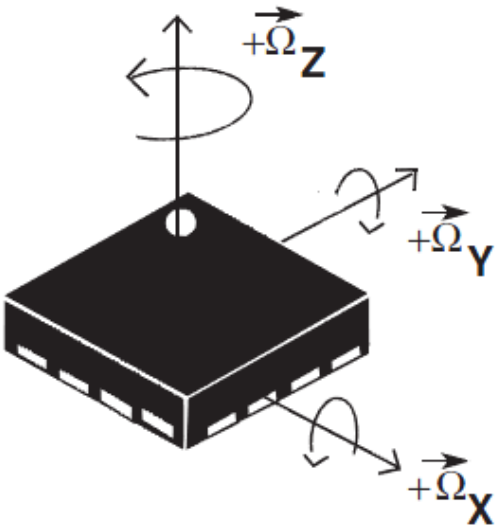

Figura 9. Dirección de velocidades angulares detectables.

Al igual que en el acelerómetro, contamos con la posibilidad de configurar la sensibilidad del sensor, teniendo la ventaja de que estos valores cambian muy poco con la temperatura, es decir, que en el rango establecido de temperatura la variación de los valores dados por el

sensor será mínima, siendo este rango de -40°C a 85°C teniendo un cambio en la sensibilidad de ±2% tal como se puede ver en la tabla 2.

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
FS	Measurement range	User-selectable		±250		dps
				±500		
				±2000		
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		
SoDr	Sensitivity change vs. temperature	From -40 °C to +85 °C		±2		%
DVoff	Digital zero-rate level	FS = 250 dps		±10		dps
		FS = 500 dps		±15		
		FS = 2000 dps		±75		
OffDr	Zero-rate level change vs. temperature	FS = 250 dps		±0.03		dps/°C
		FS = 2000 dps		±0.04		dps/°C
NL	Non linearity	Best fit straight line		0.2		% FS
Rn	Rate noise density			0.03		dps/ (√Hz
ODR	Digital output data rate			95/190/ 380/760		Hz
Top	Operating temperature range		-40		+85	°C

Tabla 2. Características mecánicas del L3GD20.

Entre las características principales de este sensor tenemos:

- Tres escalas seleccionables, 250/500/2000 dps.
- Comunicación serie I^2C/SPI.
- Formato de 16 bits de los datos de salida.
- Dos líneas digitales, una de interrupción y otra de indicar de datos listos.
- Filtro paso bajo y filtro paso bajo integrados seleccionables por el usuario.
- Rango de tensión de alimentación: 2.4V a 3.6V.
- Modos: activo y sleep.
- Incluye un sensor de temperatura.

El entorno de desarrollo Atollic TrueStudio pone a nuestra disposición un driver para la configuración y obtención de datos del sensor L3GD20, permitiéndonos su control en un lenguaje de alto nivel al estar desarrollado en el lenguaje de programación C, los detalles de la configuración del mismo se verán en el apartado *4.1.2.1 Configuración del Giroscopio.*

En la figura 10 tenemos la conexión del circuito integrado L3GD20 que incluye el sensor giroscopio y que se encuentra montado en la tarjeta STM32F3Discovery.

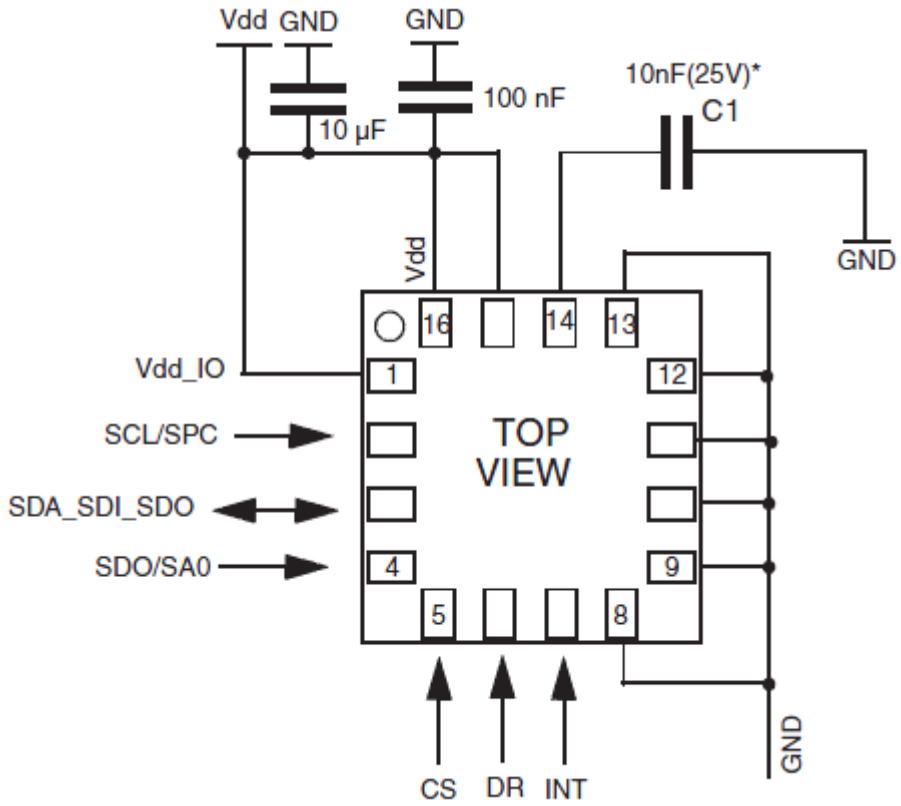

Figura 10. Conexión eléctrica del circuito integrado L3GD20.

3.6 El Hyperterminal

El programa Hyperterminal es un software que hasta Windows XP venía incorporado junto al sistema operativo, pero que desde Windows 7 ha dejado de ser incluido de serie por parte de Microsoft, por lo que se lo ha descargado de la página de la desarrolladora del programa.

Este programa nos permite establecer la comunicación del microcontrolador con el ordenador, debido a que se puede configurar la velocidad de recepción y envío de datos, el bit de paridad y el bit de stop.

Este software sólo se lo ha utilizado para las primeras comprobaciones de la comunicación entre ordenador y microcontrolador, siendo después remplazado por una aplicación de desarrollo propio, prescindiendo de esta manera de un programa de terceros.

Capítulo 4

Desarrollo del sistema de reconocimiento

La verificación de la identidad de una persona mediante el uso de la firma manuscrita 3D requiere pasar por una serie de fases que permitan al sistema tomar una decisión de la autenticidad de la firma. En el diagrama de bloques de la figura 11 se muestran cada una de las fases que un sistema de reconocimiento debe pasar antes de su validación.

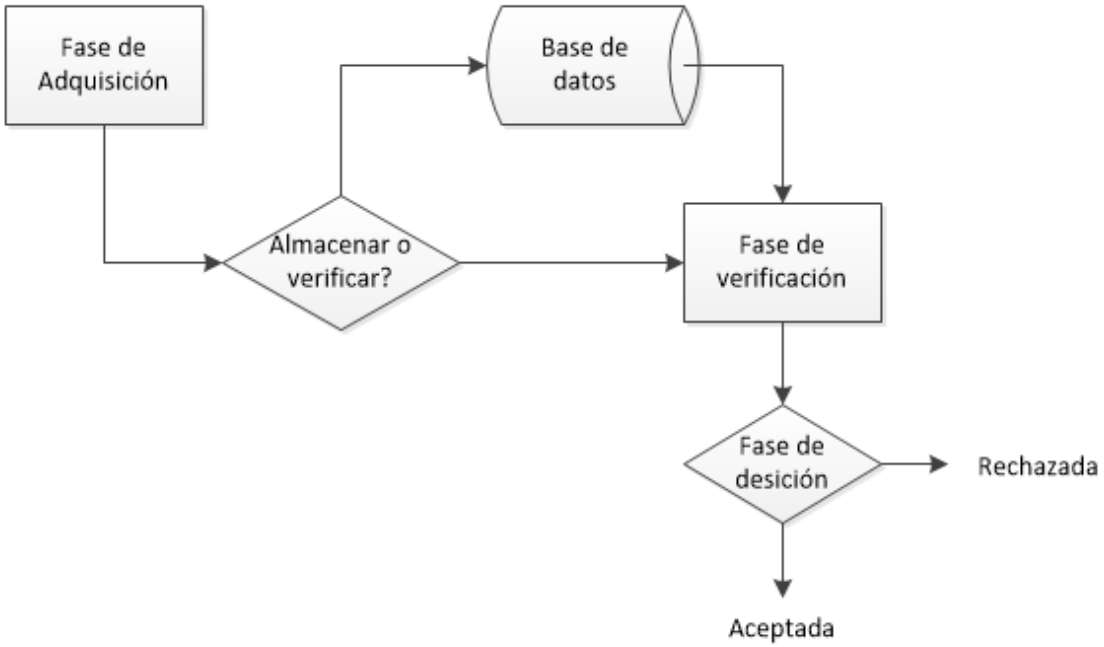

Figura 11. Diagrama de bloques del sistema de reconocimiento

Del diagrama de bloques anterior tenemos:

- **Fase de adquisición:** En esta fase se realiza la comunicación con el dispositivo y la recepción de datos enviados por el mismo.
- Luego de la fase de adquisición se dará la opción al operario de almacenar una firma en la base de datos o pasar a su verificación utilizando la firma capturada y las firmas patrones disponibles en la base de datos.
- También contamos con una base de datos, en la que se almacenarán las características de las firmas capturadas de los usuarios, que se utilizarán como firmas patrón para su posterior reconocimiento.
- **Fase de verificación:** En esta fase se implementa el algoritmo DTW y se obtienen los resultados concernientes a cada firma capturada.
- **Fase de decisión:** En esta fase, dependiendo del valor obtenido de la aplicación del algoritmo DTW, se determina la autenticidad de una firma.

Este diagrama de bloques se ha desarrollado para su implementación en el sistema, por lo que a continuación se explicarán cada uno de las aplicaciones que componen el sistema de reconocimiento y que ejecutan cada una de las fases.

4.1 Comunicación vía USB y envío de datos

Como primer paso está el conseguir la comunicación entre el ordenador y la tarjeta de desarrollo, para posteriormente enviar los datos dados por los sensores que incorpora la misma, por lo que este programa se encuentra dentro de la fase de adquisición. En la figura 12 se observa el diagrama de flujo correspondiente al programa principal, en donde se establece el orden en el que se llamarán a las funciones que configuran los distintos periféricos que se utilizan.

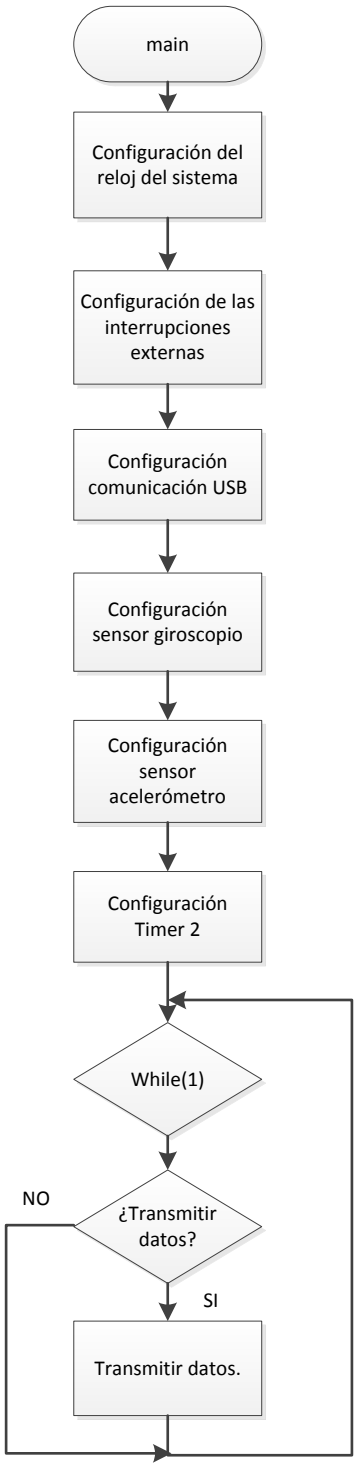

Figura 12. Diagrama de flujo correspondiente al programa principal

De la figura anterior se observa también como el programa principal tiene una espera activa mediante el empleo de un *while*, esto debido a que, realmente en la función se pregunta si se deben enviar los datos, para posteriormente llamar a la función *Transmitir_datos()*, además esta sentencia es necesaria porque de lo contrario el microcontrolador daría por finalizada la ejecución del programa.

Todas las operaciones como la temporalización y el inicio y fin de toma de datos utilizan las interrupciones, como se verá más adelante, evitando el uso de esperas activas que limiten la capacidad de multiprocesos del microcontrolador.

4.1.1 Configuración y prueba de comunicación vía USB

La comunicación entre el ordenador y el microcontrolador se realiza mediante el puerto miniUSB que incorpora la tarjeta de desarrollo STM32F3Discovery. El entorno de desarrollo Atollic TrueStudio no tiene disponible drivers para la configuración y establecimiento de la comunicación vía USB, sin embargo, el fabricante del microcontrolador, ST Electronics, pone a disposición un ejemplo de configuración y uso del puerto USB.

El ejemplo proporcionado por el fabricante ST Electronics se trata de configurar la comunicación vía USB, para posteriormente recibir un valor y devolver dicho valor por el mismo puerto.

Los ordenadores actuales ya no incluyen un puerto serie, ya que han sido remplazados por los USB al tener mayor velocidad de transferencia, por lo que para la comunicación vía USB es necesario crear una conexión virtual de puerto serie, es decir, que el ordenador pueda establecer una conexión de comunicación utilizando cualquier puerto USB disponible en el ordenador, de ahí que se conozca como Virtual Serial Port.

ST Electronics provee del controlador que habilita la función de Virtual Serial Port, permitiendo que el dispositivo sea reconocido como un elemento de recepción/transferencia conectado en un puerto serie. El ordenador utilizado cuenta con el sistema operativo Windows 8 de 64 bits.

Microsoft desde Windows Vista no permite la instalación de controladores que no figuren como una fuente fiable o que no estén firmados. Por lo que para instalar este controlador se ha seguido una serie de pasos, que se detallan a continuación:

1. Presionamos la tecla *Windows+I*con lo que nos aparecerá el panel de configuración de Windows.
2. Accedemos a la pestaña *Cambiar configuración de PC.*
3. De las opciones que aparecen en la zona lateral izquierda, seleccionamos *Uso General.*
4. Nos dirigimos a *Inicio Avanzado* y seleccionamos *Reiniciar Ahora.*
5. Una vez reiniciado el equipo, nos dará a elegir una opción de tres, en este caso seleccionamos *Solucionar problemas.*
6. Dentro de *Solucionar problemas* escogemos *Opciones avanzadas.*
7. En *Opciones Avanzadas* seleccionamos *Configuración de inicio.*
8. Posteriormente nos aparecerá una lista de las configuraciones que se podrán realizar al entrar en configuración avanzada, se reinicia el equipo.

9. Después del reinicio nos aparecerá una lista de opciones disponibles en la configuración avanzada, se selecciona la opción siete *Deshabilitar el firmado de controladores.*
10. Una vez iniciado Windows ya nos permite instalar el controlador proporcionado por ST Electronics.

Para comprobar que el ordenador reconoce a la tarjeta de desarrollo conectada al puerto USB, se debe ir a *Administrador de dispositivos* que se encuentra en el *Panel de control.* En la figura 13 se muestra como la tarjeta STM32F3Discovery ha sido reconocida por el equipo e indica el canal de comunicación asignado, el mismo tendrá que ser luego considerado para la comunicación con el hyperterminal y la programación del software de comunicación de desarrollo propio.

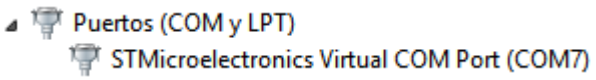

Figura 13. Tarjeta de desarrollo reconocida por el equipo.

Para la comprobación de la comunicación entre los equipos, hacemos uso del programa hyperterminal. Al abrir el mismo nos aparece un asistente de configuración, en el que seleccionamos el puerto de comunicación, en este caso el COM7, para luego establecer una serie de parámetros.

Bits por segundo: para el correcto envío y recepción de datos, tanto el ordenador como la tarjeta de desarrollo deben tener la misma velocidad de transferencia/recepción, en este caso, en el microcontrolador se encuentra configurado en 115200 bits por segundo, por lo que se ha establecido la misma velocidad en el hyperterminal. Destacar que la velocidad de transferencia en el microcontrolador es ajustable a todas las soportadas actualmente, es decir, desde 110 a 921600 bits por segundo, para establecer un valor distinto dentro del Atollic TrueStudio nos dirigimos al archivo usb_prop.c y cambiamos el valor por el deseado dentro de la estructura *linecoding.*

Bit de datos: Los datos enviados por el microcontrolador tienen un formato de 8 bits siendo este el mismo valor para los datos recibidos, por lo que en el hyperterminal se debe establecer igual valor. Al igual que la velocidad de transferencia/recepción, se puede establecer un dato distinto en la estructura *linecoding* teniendo tres opciones, 5, 6, 7 y 8 bits.

Bit de paridad: El microcontrolador está configurado para que no envíe ni espere ningún bit de paridad, por lo que en el hyperterminal también se debe establecer la no necesidad de un bit de paridad. Este parámetro también es configurable, están incluidos en la misma estructura que los anteriores.

Bits de parada: En este caso se ha establecido un bit de parada, este valor es modificable en la estructura *linecoding*.

Una vez configurados los parámetros necesarios para la comunicación se procede a realizar las pruebas necesarias. Por lo que, con la placa conectada al ordenador, comprobamos como al presionar una tecla, el microcontrolador recibe el dato y al instante lo devuelve, apareciendo el carácter de la tecla pulsada en la ventana del hyperterminal, de esta forma se ha verificado la correcta comunicación entre ordenador y microcontrolador.

4.1.2 Configuración de sensores

Utilizando los drivers que pone a disposición Atollic TrueStudio, se ha configurado los distintos parámetros de los sensores, utilizando un lenguaje de alto nivel al estar escrito en C.

La configuración se encuentra en funciones fuera de la rutina principal *main.c,* para tener un mayor nivel de organización. Desde el programa principal son llamadas las funciones *Gyro_Config y Acc_Config* para realizar sus respectivas configuraciones.

4.1.2.1 Configuración del Giroscopio

La posibilidad de configurar la sensibilidad del sensor giroscópico, nos permite establecer una que se adapte mejor a las necesidades del proyecto. El sensor giroscópico sólo permite ser configurado por tres valores predefinidos, es decir, no permite que el usuario pueda seleccionar un valor que no sea uno de los establecidos, siendo estos: 250 dps, 500 dps y 2000 dps.

Para determinar el valor de la sensibilidad, se han realizado dos pruebas con el dispositivo que integra este sensor. La primera prueba ha consistido en dejar la tarjeta de desarrollo en reposo y la segunda prueba ha consistido en realizar movimientos lentos y rápidos con la tarjeta de desarrollo.

De estas pruebas se ha podido observar que:

- Con una sensibilidad de 250 dps el sensor sólo es capaz de detectar velocidades angulares grandes, es decir, pequeños movimientos y movimientos intermedios eran desapercibidos por el sensor.
- Con una sensibilidad de 2000 dps obtenemos el efecto contrario, es decir, que el más pequeño movimiento era detectado, incluso en reposo se obtenían valores superiores a la unidad, comprobándose la existencia de ruido.
- Con una sensibilidad de 500 dps, se consigue un equilibrio entre la detección de pequeños y grandes movimientos, por lo que esta sensibilidad es la que se establece en el sensor para su uso.

Se establece la actualización de datos en el buffer de salida en modo continuo y que los datos obtenidos sean respecto a los tres ejes, ya que también el sensor nos permite seleccionar la obtención de datos de un solo eje, pero en este caso necesitamos la velocidad angular respecto a cada eje.

4.1.2.2 Configuración del Acelerómetro

En el acelerómetro se ha habilitado la obtención de datos respecto a los tres ejes, se ha establecido una sensibilidad de ±2g y que la obtención de datos se realice de forma continua, ya que lo que se desea es obtener las aceleraciones durante la realización de la firma y no un dato puntual.

La elección de la sensibilidad del acelerómetro se la ha realizado siguiendo el mismo criterio que para el giroscopio, es decir, que el sensor sea capaz de detectar la mayoría de los movimientos durante la realización de la firma, de ahí que no se haya elegido una sensibilidad mayor, ya que detectaría pequeños movimientos, pudiendo ser considerados ruido, lo que conllevaría a crear dificultades en el proceso de verificación.

Una vez configurado los sensores, el microcontrolador se encuentra en disposición de poder leer los buffer de datos de cada uno de ellos para su posterior conversión a caracteres y envío al ordenador.

4.1.3 Captura y envío de datos

Para establecer la frecuencia con la que se obtienen los datos entregados por los sensores, se ha obtenido el tiempo que tarda el microcontrolador en enviar los datos. Para ello se ha habilitado un pin del microcontrolador que se mantiene en nivel lógico 1 cuando se encuentra enviando datos y en nivel lógico 0 cuando está realizando cualquier otra operación.

Mediante un osciloscopio se ha obtenido la señal cuadrada que entrega ese pin y de donde se ha determinado el tiempo de envío de datos por parte del microcontrolador, siendo el mismo de 8.33 ms.

Por lo que para la obtención de datos se utiliza el timer 2, configurado para que cada 10 ms se produzca una interrupción durante la cual se leen los buffer de datos de cada sensor y se almacenan en una variable.

Al producirse una captura de datos cada 10ms, se obtienen 100 capturas por segundo, pero en cada una de ellas se consigue el valor de la aceleración y la velocidad angular respecto de cada eje, por lo que cada captura está compuesta por seis datos, que son almacenados en una variable tipo array. De esta forma se obtienen 6 datos cada 10 ms, que expresados en segundos serán 600 datos por segundo.

Para conseguir este período de muestreo se ha configurado al timer 2, mediante la función *Tim_Config(),* estableciendo el valor del prescaler en 7200 y un valor de 100 en el periodo del mismo, se determina un divisor de reloj igual a uno y sin repetición de la cuenta, también se habilita las interrupciones por actualización del timer. Si el reloj del microcontrolador se encuentra establecido en 72 MHz, entonces tenemos:

$$Frecuencia\ del\ Timer = \frac{Reloj\ del\ sistema}{Prescaler\ del\ timer} = \frac{72 \cdot 10^6}{7200} = 10\ KHz \qquad [2]$$

$$Tick\ del\ Timer = \frac{1}{Frecuencia\ del\ Timer} = \frac{1}{10000} = 100\ \mu s \qquad [3]$$

Por lo que el timer incrementa su cuenta en una unidad cada 100 us, pero la captura se realiza cada cien cuentas del timer, por lo que:

$$Captura = \ Tick\ del\ Timer \cdot Periodo = 100\ \mu s \cdot 100 = 10\ ms \qquad [4]$$

Obteniendo de esta forma un período de muestreo de 10 ms en la captura de datos. En la figura 14 se observa el diagrama de flujo correspondiente a la rutina de atención a la interrupción del timer 2, en la misma se aprecia la obtención de datos y la puesta a uno de la variable que habilita la transmisión de datos.

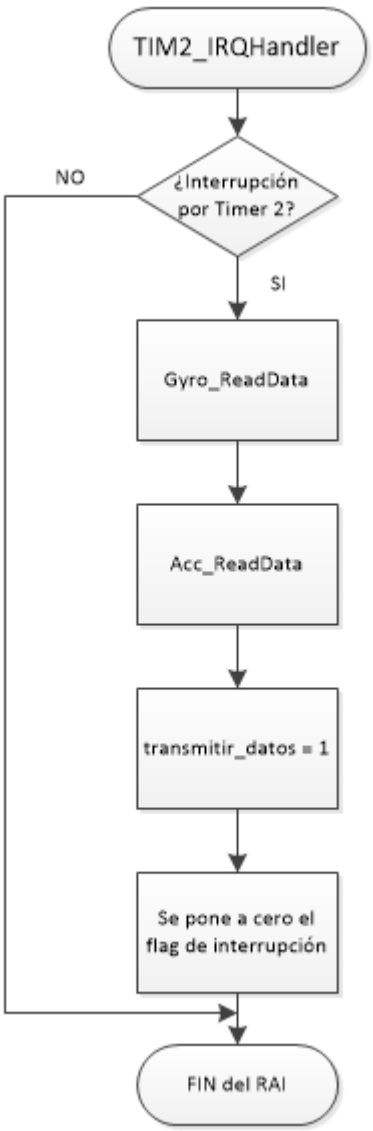

Figura 14. Diagrama de flujo del RAI correspondiente al Timer 2

Para evitar capturar valores cuando no se está realizando la firma y que los mismos sean considerados en el proceso de verificación alterando el resultado, se ha configurado el pulsador USER conectado al puerto A, pin cero del microcontrolador e incorporado en la

tarjeta de desarrollo, como una fuente de interrupción externa, de esta manera el timer 2 será habilitado cuando el pulsador USER sea accionado, debido a que la rutina de atención a la interrupción de la fuente externa habilita el inicio de cuenta del temporizador.

De la misma manera al finalizar la realización de la firma se accionará el pulsador para indicar al microcontrolador la finalización de la misma, en este caso se deshabilita el timer 2 y por ende la captura y envío de datos. En la figura 15 se muestra el diagrama de flujo correspondiente a la rutina de atención a la interrupción del pulsador USER que realiza las operaciones antes expuestas.

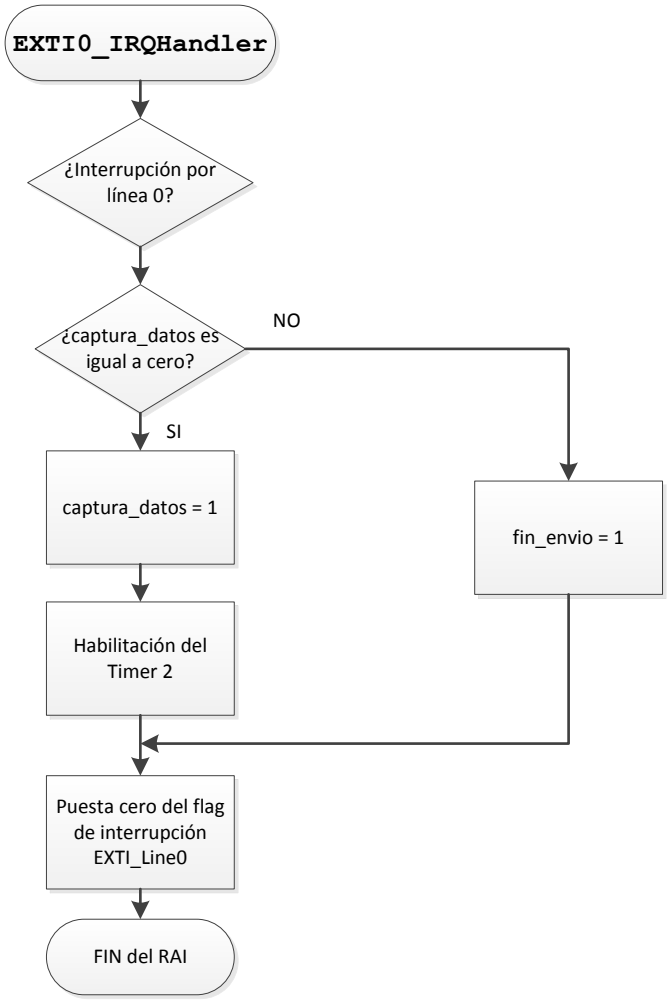

Figura 15.Diagrama de flujo del RAI correspondiente a la EXTI0

El envío de datos se realiza por cada muestra obtenida y no cuando se ha terminado de realizar la firma, esto debido a la limitación que nos impone la memoria disponible en el microcontrolador, además de que los valores tipo *float* ocupan un mayor espacio que los tipo *int*, y se trabaja con los primeros, de ahí que los datos se envíen de forma continua.

Para el envío de datos se procesa cada uno de los valores obtenidos, ya que para la recepción de los mismos por parte del ordenador es necesario convertir los valores tipo *int* a tipo *char*, empleando el código ASCII, por lo que cada uno de los valores disponibles en el array primero son convertidos a caracteres y luego enviados.

Para el envío de nuevos datos se debe asegurar que el dato anterior ya ha sido enviado, es decir que el periférico no se encuentra ocupado, por ello se verifica el fin de envío, antes de escribir un nuevo valor en el buffer de salida. En la figura 16 se muestra el diagrama de flujo correspondiente al procesado y envío de datos.

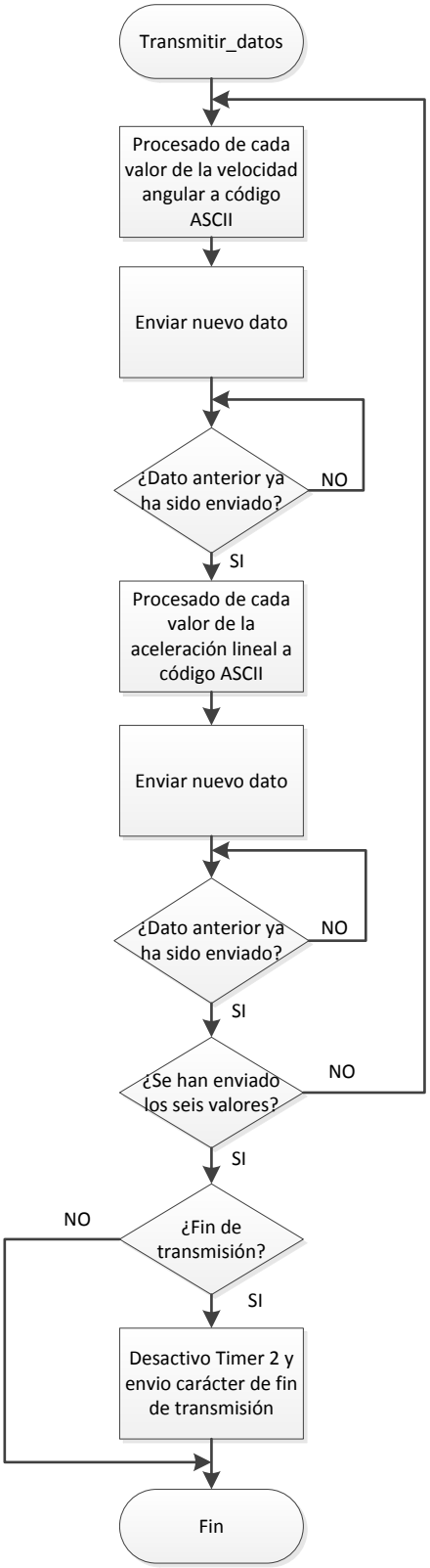

Figura 16. Diagrama de flujo correspondiente a la función de envío de datos

Por último, para evitar problemas en el envío de los datos, se ha comprobado que la operación de transmisión se realiza en un tiempo menor al tiempo de captura, es decir, el tiempo que le toma al microcontrolador realizar la lectura de los buffers, la conversión a código ASCII y el posterior envío de los datos vía USB es inferior a 10 ms que es el período de captura, evitando así que el ordenador reciba datos incompletos y que afecten en el proceso de verificación.

4.1.4 Estructura del programa

Una vez expuestas las configuraciones necesarias y el establecimiento de la comunicación, en la tabla 3 se describen las funciones que componen el programa principal.

Función	Descripción
Transmitir_datos()	Realiza el procesamiento de datos en código ASCII para su posterior envío a través del puerto USB.
EXTI_Config()	Configuración del pin 0 del puerto A como fuente de interrupción externa. Habilita las interrupciones y establece la prioridad.
USB_Config()	Configuración del puerto USB, habilita los relojes del sistema correspondientes e interrupciones. Inicia el funcionamiento de la comunicación USB.
Gyro_Config()	Configuración del sensor giroscópico, estableciendo sus parámetros. Inicialización.
Acc_Config()	Configuración del sensor correspondiente al acelerómetro, estableciendo sus parámetros. Inicialización.
Tim_Config()	Configuración del temporizador, en donde se establece el tiempo de captura de datos. Habilita las interrupciones correspondientes a este periférico.
TIM2_IRQHandler()	Es la rutina de atención a la interrupción asociada al Timer 2. Obtiene los datos de los sensores y habilita el envío de los mismos al ordenador.
EXTI0_IRQHandler()	Rutina de atención a la interrupción correspondiente a la fuente de interrupción externa. Habilita y deshabilita la captura de datos según corresponda.
Gyro_ReadData(float* pfData)	Lee los datos dados por el sensor del giroscopio, almacenándolos en una variable tipo array.
Acc_ReadData(float* pfData)	Lee los datos dados por el sensor del acelerómetro y los almacena en una variable tipo array.
L3GD20_TIMEOUT_UserCallback()	En caso de problemas en la comunicación con los sensores, el sistema bloquea la comunicación con los mismos y evita todo proceso. Estas funciones entran en un bucle infinito.
LSM303DLHC_TIMEOUT_UserCallback()	

Tabla 3. Funciones del programa principal y su descripción.

En la tabla 4 se detallan las variables y definiciones empleadas en el programa principal exponiendo sus funciones dentro del mismo.

		Descripción
Variables	***GyroBuffer[3]***	Variable tipo array en la que se almacena los tres valores dados por el giroscopio en cada proceso de lectura.
	AccBuffer[3]	Variable tipo array en la que se almacena los tres valores dados por el acelerómetro en cada proceso de lectura.
	Captura_datos	Dependiendo de su valor se utiliza para habilitar o deshabilitar la captura de datos.
	Packet_sent	Si su valor es '1' los datos aún no han terminado de ser enviados, si es '0' el proceso de envío ha terminado. Se utiliza para evitar sobrescribir el buffer de envío del USB y tener pérdidas de datos.
	Ncentenas	En estas variables se almacenan cada dígito de los valores obtenidos de los sensores, estando estos codificados en código ASCII para su posterior envío.
	Ndecenas	
	nunidades	
	ndecimales	
	transmitir_datos	Habilita o deshabilita la transmisión de datos.
	fin_envio	Indica si los datos han terminado de ser enviados, permitiendo deshabilitar la captura de los mismos y envía al ordenador el carácter indicativo de fin de envío.
	Receive_Buffer[64]	Variable tipo array en la que se agrupan los valores en código ASCII para el envío continúo por el puerto USB.
	Receive_length	Indica el número de caracteres a enviar.
Definiciones	***PI***	Establece el valor de PI.
	LSM_Acc_Sensitivity_2g	Establecen las distintas sensibilidades disponibles en el acelerómetro
	LSM_Acc_Sensitivity_4g	
	LSM_Acc_Sensitivity_8g	
	LSM_Acc_Sensitivity_16g	

Tabla 4.Variables y definiciones en el programa principal.

4.2 Desarrollo del software de recepción y almacenamiento de datos

Para la comprobación del correcto funcionamiento del envío de datos por parte del microcontrolador al ordenador se utilizó el programa hyperterminal, pero teniendo como objetivo prescindir de un programa de terceros, se ha desarrollado un software propio en Visual Studio 2012 mediante el lenguaje de programación Visual Basic, que permite de una forma sencilla acceder a los puertos del sistema para su escritura y lectura. Este programa al igual que el desarrollado para el microcontrolador se encuentra dentro de la fase de

adquisición de datos, además también implementa el bloque que da la opción al usuario de almacenar o verificar la firma capturada.

4.2.1 Funcionamiento del programa

La estructura de la función principal *main* se muestra en la figura 17 en donde se expone el diagrama de flujo de la misma. Como se observa ejecuta las funciones necesarias para la recepción de datos provenientes del microcontrolador.

Al ejecutar la aplicación se visualizará por la ventana de comandos dos opciones. La primera se seleccionará si se desea iniciar el proceso de ejecución de la firma 3D, por lo que ejecutará la función *Recepcion_datos()*, para posteriormente dar a elegir entre almacenar los datos respectivos a esa firma capturada o iniciar el proceso de verificación de la firma, llamando a las funciones *Almacenamiento_datos() y Verificar_datos()* respectivamente.

En cuanto a la segunda opción permite eliminar usuarios de la base de datos, en este caso llama a la función *Eliminar_usuario()*.

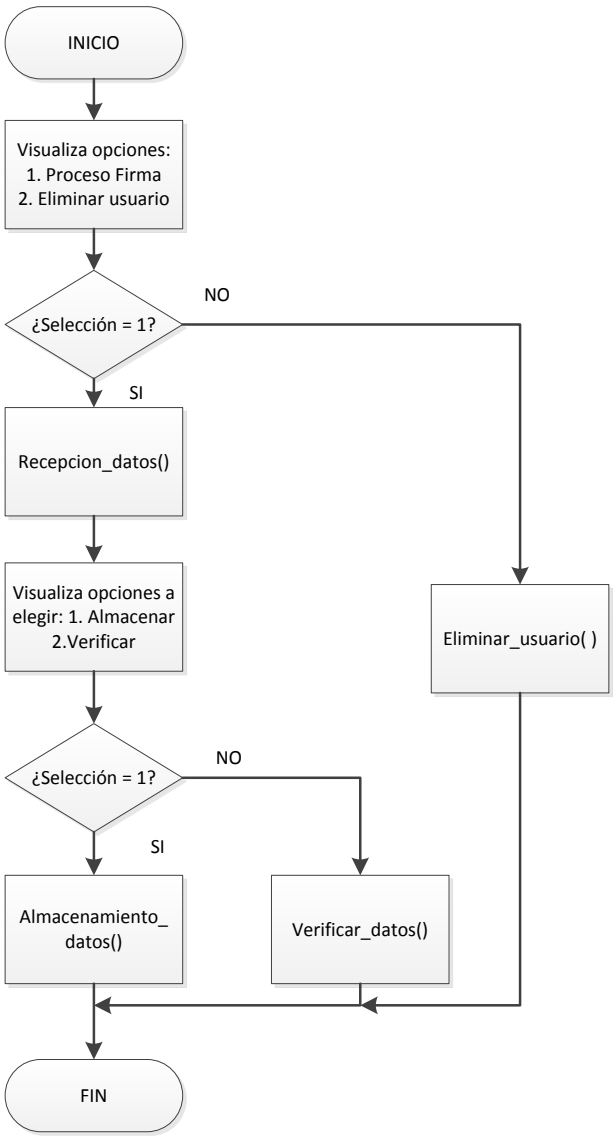

Figura 17. Diagrama de flujo de la función *main().*

En la figura 18 se observa el diagrama de flujo correspondiente a la función *Recepción_datos()*. En la misma, se comprueba que el dispositivo de captura esté conectado al ordenador por USB y se espera que el usuario inicie el envío de los valores obtenidos de los sensores de la tarjeta de desarrollo, para ello, como se ha expuesto en el apartado *4.1.3 Captura y envío de datos,* se debe accionar el pulsador USER, momento en el cual se mostrarán por la ventana de la consola los valores recibidos.

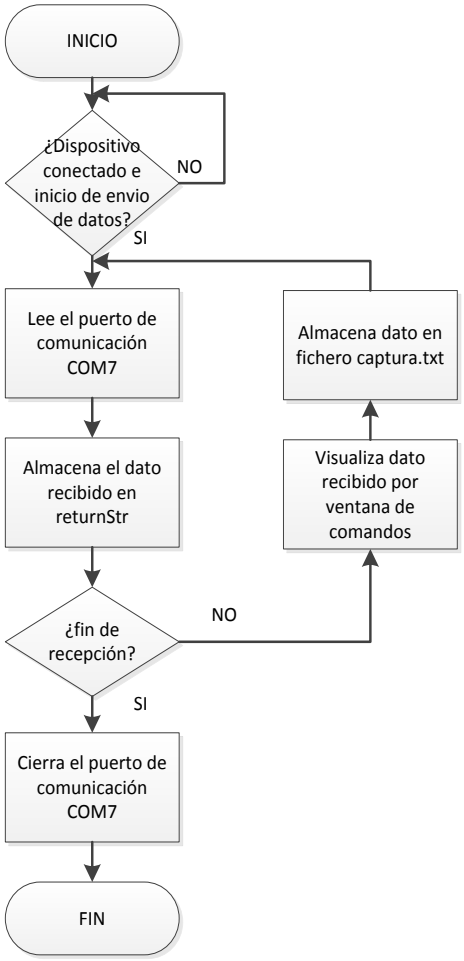

Figura 18. Diagrama de flujo de la función Recepción_datos().

Una vez iniciada la recepción de datos, se lee el puerto de comunicación al que se encuentra conectado el dispositivo de captura (STM32F3Discovery), siendo en este caso el puerto COM7 implementado en el ordenador que se ha utilizado para el desarrollo del programa, para posteriormente almacenar el dato recibido en la variable *returnStr*.

A continuación se verifica si el dato almacenado en la variable *returnStr* corresponde con el carácter que indica el fin de recepción, siendo en este caso la letra 's', de no darse el caso se visualiza el dato por pantalla para a continuación almacenarlo en el archivo *captura.txt*, repitiendo el proceso de recepción de datos. En caso de recibir el carácter de fin de recepción, se cierra el puerto de comunicación y se finaliza la ejecución de la función regresando al programa principal.

En la figura 19 se expone el diagrama de flujo correspondiente a la función *Almacenamiento_datos().* Esta función solicita la introducción del nombre de usuario que se asociará a los datos recibidos, para a continuación leer el fichero *usuarios.txt* del cual se obtiene el número de firmas almacenadas en la base de datos, antes de cerrar el mismo se actualiza incrementando en una unidad para establecer el nuevo tamaño de la base de datos.

Seguidamente se copia el fichero *captura.txt* en uno nuevo denominado *datos_x.txt* en donde 'x' corresponde al número de datos disponible en la base de datos que se le asigna. Por último se cierran los ficheros y se regresa a la función principal.

Indicar que el fichero *captura.txt* es un fichero que almacena datos de forma temporal, debido a esto, para el almacenamiento se copian los valores a un nuevo archivo que será el que formará parte de la base de datos.

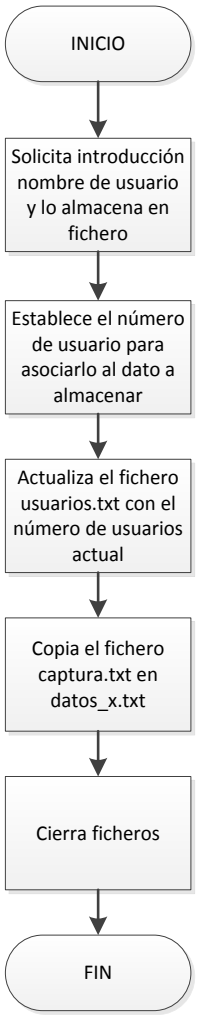

Figura 19. Diagrama de flujo de la función Almacenamiento_datos().

Otra función que forma parte de la aplicación de recepción de datos es la función *Verificar_datos()*, cuyo diagrama de flujo se muestra en la figura 20. Al ejecutar esta función nos solicita la introducción del nombre de usuario al que se va a verificar la firma, como es necesario recorrer toda la base de datos en busca de los cinco ficheros que almacenan las cinco muestras que se utilizan como firmas patrones, primero se lee la longitud de la base de

datos, cuyo valor es el almacenado en el fichero *usuarios.txt* y se lo asigna a la variable *usuarios_reg*.

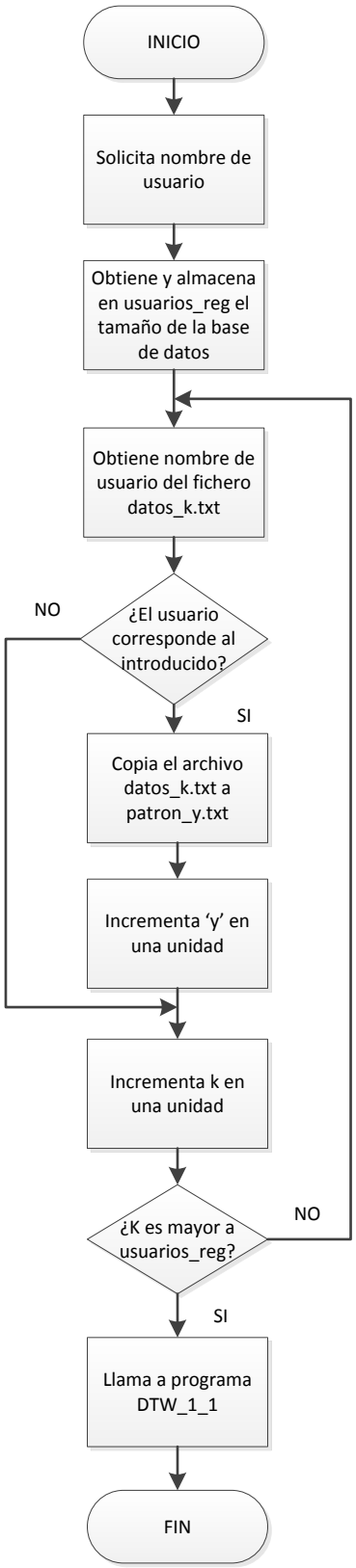

Figura 20. Diagrama de flujo de la función Verificar_datos().

Posteriormente se lee el fichero *datos_k.txt* en donde 'k' corresponde al número de fichero actual al que se está accediendo, del mismo se obtiene el nombre de usuario para a continuación compararlo con el introducido; en el caso de que sea el buscado se copia el fichero *datos_k.txt* al fichero *patrón_y.txt* en donde 'y' es una variable que establece el número de patrón disponible, siempre empezando por el número 1 y terminando en el número 5.

Seguidamente se incrementa en una unidad la variable 'k' y se verifica si es mayor a la variable *usuarios_reg* que contiene la longitud de la base de datos, de no darse el caso se accede al siguiente fichero y se repite el proceso, consiguiendo así recorrer toda la base de datos. Por último, tenemos la función *Eliminar_usuario(),* su diagrama de flujo se muestra en la figura 21.

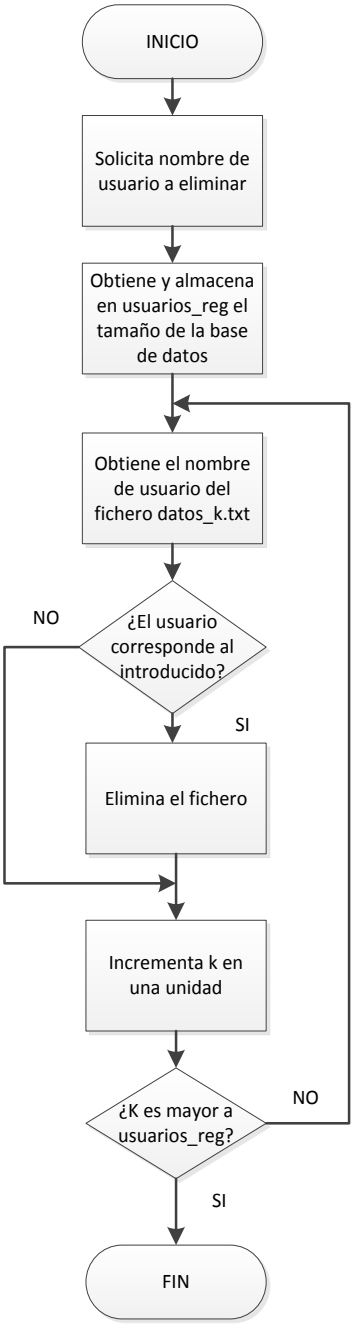

Figura 21. Diagrama de flujo de la función *Eliminar_usuario()*.

La función *Eliminar_usuario()* nos permite borrar de la base de datos un usuario determinado. Para ello, primero se solicita la introducción del nombre del usuario a eliminar, el programa busca en la base de datos los ficheros que contienen los datos correspondientes al usuario introducido.

Una vez localizado el fichero, la aplicación muestra un mensaje advirtiendo de que el archivo será eliminado y pregunta si desea continuar con la operación. Una vez eliminado el usuario, se visualizará por la ventana de comandos un mensaje, que pueden darse dos casos:

- En caso de que se ha finalizado el proceso de eliminación de usuario correctamente mostrará un mensaje de *usuario eliminado correctamente*.
- En caso de que el usuario introducido no existe en la base de datos, mostrará el mensaje e*l usuario introducido no se encuentra disponible en la base de datos*. Este caso también es aplicable a la función *Verificar_datos().*

Este programa además de comunicarse por el puerto USB con el microcontrolador para la recepción de datos y almacenamiento de los mismos, en el caso de la verificación, también llama a un segundo programa que implementa el algoritmo, el mismo también es una aplicación de desarrollo propio, en donde se hace uso de una librería para la implementación del Dynamic Time Warping.

4.2.2 Estructura del programa

El programa de recepción y almacenamiento de datos se ha estructurado mediante tres funciones, además de la función principal. De esta forma se ha conseguido una organización óptima, ya que cada función realiza una tarea específica. Las funciones se detallan en la tabla 5 y en la tabla 6 se muestran las variables utilizadas para su desarrollo.

		Descripción
Funciones	***Main()***	Ejecuta las funciones en un orden establecido para el correcto funcionamiento del programa
	Recepcion_datos()	Recibe los datos provenientes del microcontrolador y los almacena en un fichero denominado captura.txt
	Almacenamiento_datos()	Solicita el nombre de usuario asociado a los datos de la firma recibidos y los almacena en la base de datos.
	Verificar_datos()	Solicita el nombre de usuario del que se desea verificar la firma, una vez introducido busca en la base de datos los ficheros que contiene las firmas patrón correspondientes al usuario para posteriormente llamar al programa de aplicación del algoritmo y resultados.
	Eliminar_usuario()	Solicita la introducción del usuario a eliminar de la base de datos. Elimina los ficheros que almacenan los datos de las firmas correspondientes al usuario introducido.

Tabla 5. Funciones empleadas en el programa de recepción y almacenamiento de datos.

		Descripción
Variables	***usuarios_reg***	Se utiliza como variable auxiliar para almacenar temporalmente el número de usuarios registrados en la base de datos, es requerida posteriormente en la función *Verificar_datos().*
	fichero	Determina el nombre del fichero a acceder.
	usr	Almacena el nombre de usuario del fichero actual para posteriormente compararlo con el nombre de usuario introducido y determinar si es una firma patrón de ese usuario
	patrones	Determinan el nombre de fichero en el que se copiará los datos de la firma patrón.
	ptr	
	arch_encontrado	Variable en la que se almacena el valor de la comparación entre nombre de usuario del fichero y el nombre de usuario introducido, si es 0 se trata de una firma patrón, si es 1 los datos son de otro usuario.
	nuevo_usuario	Utilizada para determinar el número de usuario y que complementa al nombre del fichero a almacenar, por ejemplo, datos_**10.**
	returnStr	Almacena el dato recibido vía USB para posteriormente ser escrito en el fichero *captura.txt.*
	selección	Almacena la selección realizada por el usuario, al preguntar sobre si desea almacenar o verificar la firma.
	strLine	Guarda la línea de datos leída desde el fichero.
	objStreamReaderUsuario	Punteros a ficheros para escritura y lectura
	fs	
	path	Rutas de almacenamiento
	almacenar	

Tabla 6. Variables utilizadas en el programa de recepción y almacenamiento de datos.

4.3 Implementación del algoritmo DTW y fase de decisión.

Para la implementación del algoritmo DTW se ha desarrollado un programa en Visual Studio 2012 utilizando el lenguaje de programación C-Sharp. En este apartado también se resaltan las propiedades del Dynamic Time Warping que lo hacen más robusto frente a otro algoritmo como la distancia Euclídea. Además también se han analizado una serie de librerías del DTW disponibles actualmente, eligiendo aquella que se adapte mejor a nuestro proyecto. Posteriormente se explican las principales funciones que componen el programa.

4.3.1 Ventajas del Dynamic Time Warping frente a la distancia Euclídea.

El algoritmo DTW utiliza un método determinado, como paso previo, para medir la distancia, pudiendo ser el mismo, la distancia Euclídea, la distancia Manhattan o la distancia Squared Euclídea, para posteriormente devolver el valor de la distancia medida entre dos curvas, siendo este el producto del cuadrado de la distancia entre el resultado de la raíz cuadrada de la

suma de productos del número de puntos que componen cada curva, obteniéndose como resultado un valor de DTW normalizado en longitud [14].

Si utilizamos sólo la distancia Euclídea, podría ocasionar problemas en la fase de decisión, esto debido a que este método sólo mide la distancia entre puntos en el eje vertical de dos curvas, sin considerar la desviación en el tiempo [14], en la figura 22 se puede apreciar un ejemplo de la medida de la distancia entre dos series temporales por distancia Euclídea.

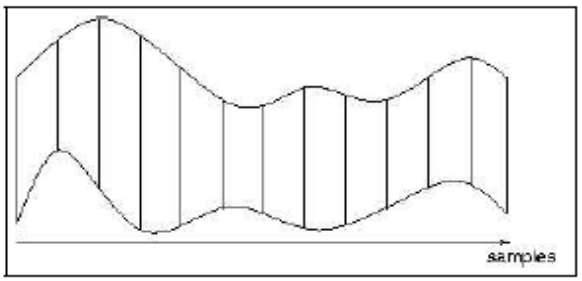

Figura 22. Resultado de aplicar la distancia Euclídea.

El DTW soluciona el problema de la distancia Euclídea, ya que localiza los puntos similares correspondientes en la curva 1 y en la curva 2, para posteriormente utilizando la distancia Euclídea medir la misma entre los puntos señalados, encontrando una alineación optima entre las dos series [11]. En la figura 23 se puede apreciar el resultado de aplicar el algoritmo DTW.

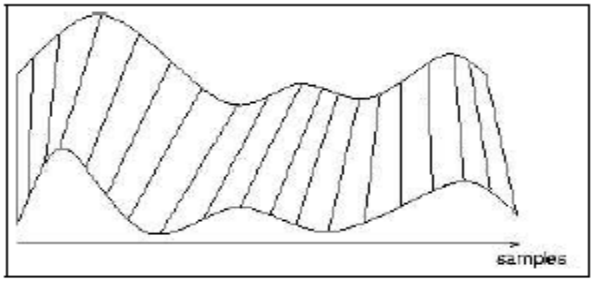

Figura 23. Resultado de aplicar el DTW

Se observa como el Dynamic Time Warping (DTW) ha medido distancias entre puntos en los que las curvas presentan similitud, así vemos como los puntos más altos de la curva 1 se han asociado con los puntos más altos de la curva 2 para obtener su distancia, e igual con los demás puntos, de esta manera se considera la desviación temporal.

La utilización del Dynamic Time Warping puede tener como inconveniente el elevado coste computacional del que pueda requerir, pero dicho coste depende de la cantidad de datos que se va a procesar con el algoritmo, no existiendo inconveniente alguno si el número de valores no es muy grande, por lo que para el desarrollo del proyecto la utilización del DTW no supone ningún inconveniente para el ordenador que se empleará para su aplicación.

4.3.2 Elección de la librería de implementación del algoritmo DTW

En la actualidad existen una gran variedad de librerías disponibles para la implementación del algoritmo DTW, tanto de código abierto como de código propietario. Para evitar infringir

patentes que tengan asociadas el código propietario, se ha considerado solo las librerías de código abierto, entre las que tenemos:

- ***Librería Ibimproved***: implementa el algoritmo del vecino más próximo para la aplicación del Dynamic Time Warping, la misma se encuentra desarrollada en lenguaje de programación C++.
- ***Librería FastDTW***: es un driver desarrollado en lenguaje Java que implementa el algoritmo DTW, hace uso de una mejora de dicho algoritmo para aumentar la velocidad del mismo.
- ***Librería Package R DTW***: implementa variantes conocidas del algoritmo DTW, incluyendo restricciones y sub-cadenas coincidentes.
- ***Librería mlpy***: driver desarrollado en lenguaje de programación Python que implementa el DTW.
- ***Librería JavaML***: implementa la máquina de aprendizaje del DTW.
- ***Librería ndtw***: desarrollada en lenguaje de programación C-Sharp (C#), implementa el algoritmo DTW con una variedad de opciones.
- ***Librería Sketch-a-Char***: escrito en JavaScript, implementa el DTW.
- ***Librería MatchBox***: implementa el DTW, orientado más a señales de audio.

La elección de la librería se ha realizado considerando el lenguaje de programación con el que se ha desarrollado y la compatibilidad con la aplicación que estamos desarrollando, por lo que la librería que se utiliza para el desarrollo de este proyecto es la *ndtw*, al estar escrita en lenguaje C-Sharp y poseer varias opciones de configuración.

4.3.3 La Librería ndtw

Utilizando la librería ndtw desarrollada por Darjan Oblaky con licencia MIT, siendo la misma concedida por el Instituto Tecnológico de Massachusetts, permitiéndonos un uso variado de la misma, tales como: usar, copiar, modificar, integrar con otro software, publicar, sub-licenciar o vender copias del software, y además permitir a las personas a las que se les entregue el software hacer lo mismo.

Siempre con la condición de que la nota de copyright y la parte de los derechos se incluya en todas las copias o partes sustanciales del software. En nuestro caso hacemos uso de esta librería para incluirla como parte de nuestro programa.

La librería *ndtw* desarrollada en lenguaje C-Sharp (C#) cuenta con las siguientes características:

- La utilización de dos o más características.
- Pre-procesamiento de datos como: el centrado, normalización y estandarización.
- La asignación de peso a las variables como una función opcional.
- Medida de las distancias por: Manhattan, Euclídea, Squared Euclídea.

Por lo que esta librería nos permite implementar distintos tipos de medida de la distancia o desviación, así como un pre-procesamiento de los datos. El algoritmo DTW desarrollado en esta librería, devuelve un valor de la comparación de dos señales o como en nuestro caso comparación de firmas, dependiendo del nivel de semejanza, por lo que para firmas muy

similares el valor DTW devuelto es bajo mientras que en firmas distintas el valor DTW devuelto es mayor, comportamiento que tendremos en cuenta en la fase de decisión.

4.3.4 Funcionamiento del programa

En la figura 24 se observa el diagrama de flujo correspondiente al programa principal, en donde se ejecutan las diferentes funciones para la aplicación del algoritmo. En el mismo además se calcula el coste normalizado, siendo este necesario para obtener un valor de DTW normalizado por longitud y que se encuentra establecido por el número de datos de los que se disponen por característica.

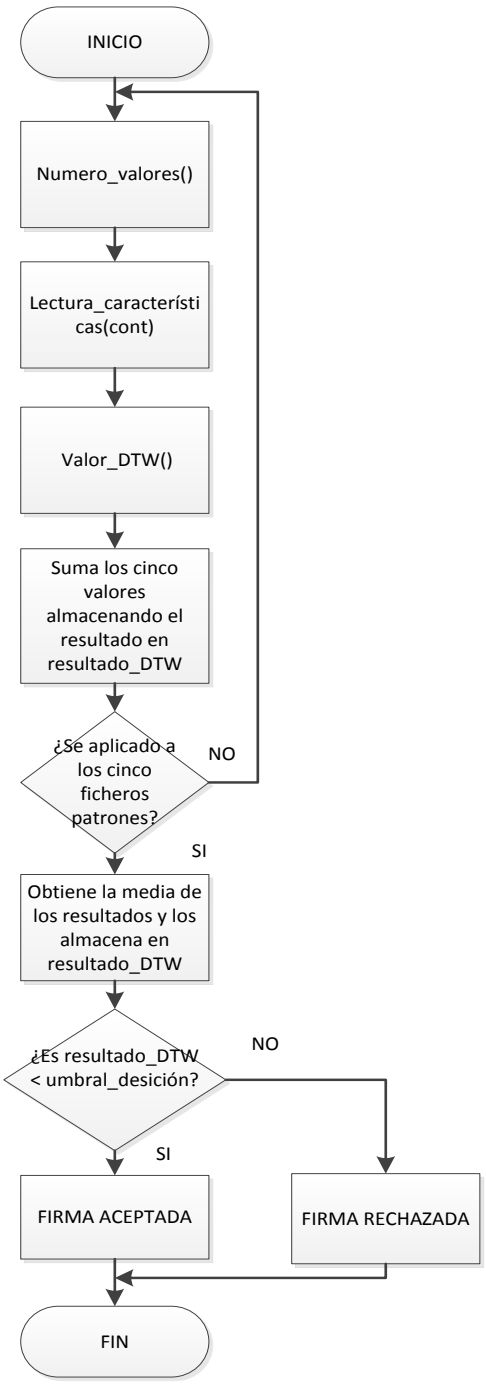

Figura 24. Diagrama de flujo del programa principal de implementación DTW.

El programa principal asegura que el algoritmo sea aplicado a las cinco firmas patrones de las que se dispone, sumando el resultado devuelto por la función *Valor_DTW()* para cada fichero, una vez finalizado este proceso se obtiene la media de los resultados de la aplicación del DTW, obteniendo así el valor final que es comparado con el umbral de decisión, por lo que si el resultado es menor a dicho umbral la firma es aceptada en caso contrario la firma es rechazada, finalizando de esta manera el proceso de reconocimiento de la firma manuscrita online 3D.

En la figura 25 se observa el diagrama de flujo correspondiente a la función *Número_valores,* que devuelve al programa principal el número de datos de los que dispone cada característica.

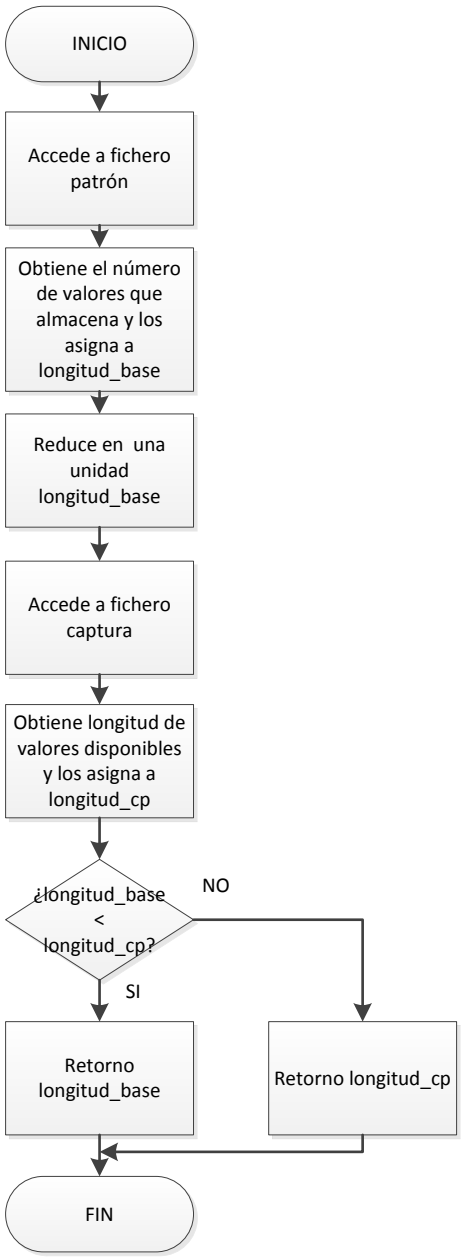

Figura 25. Diagrama de flujo de la función *Número_valores()*

En el programa del microcontrolador se ha asegurado que todas las características tengan el mismo número de valores, por lo que es suficiente realizar una única lectura del fichero que

contiene los datos de la firma y dividirlo entre seis, ya que se cuenta con seis características por firma y así obtener la longitud por cada característica que se dispone.

Al número de valores disponibles contenidos en un fichero se le resta una unidad, esto debido a que al final del mismo se encuentra el nombre del usuario, de esta forma se evita contabilizar ese dato como valor disponible.

En la figura 26 se expone el diagrama de flujo correspondiente a la función *Lectura_características,* la misma realiza la lectura del fichero que contiene los datos de la firma, almacenándolos en variables tipo array, clasificando los valores en las seis características de las que se disponen.

Esta operación se realiza tanto para la firma patrón como para la firma capturada, ya que la comparación se realiza entre ambas. Una vez obtenidos los datos de los archivos y almacenados en variables nos encontramos en condición de aplicar el algoritmo DTW.

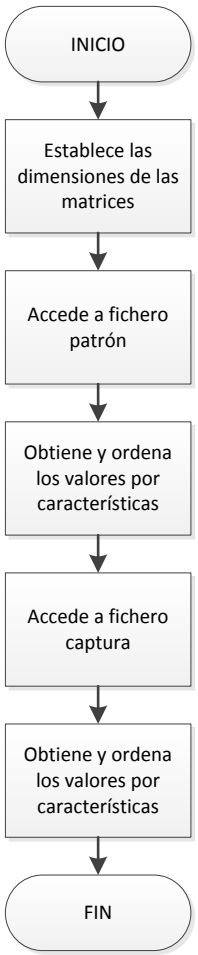

Figura 26. Diagrama de flujo de la función Lectura_características().

Posteriormente la función *Valor_DTW(),* hace uso de las variables en donde se han almacenado los valores de las características para aplicar el algoritmo y visualizar los resultados por la ventana de comando.

En la figura 27 se muestra el diagrama de flujo correspondiente a la función *Valor_DTW()*. Como se puede observar se obtiene el valor del DTW aplicado a cada una de las cinco firmas patrones de las que se disponen, almacenando cada resultado parcial en variables que posteriormente se sumarán y se obtendrá la media de los cinco resultados parciales.

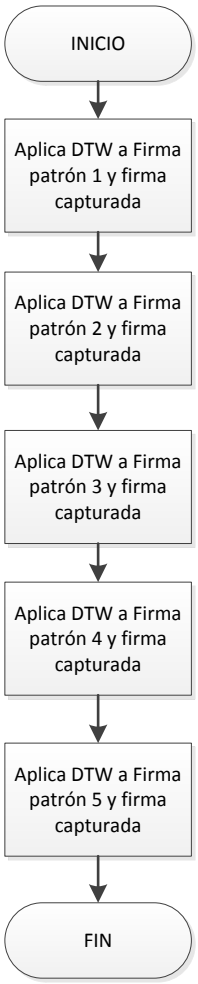

Figura 27. Diagrama de flujo de la función Valor_DTW().

Al ejecutar el programa nos aparecerá una ventana con los resultados, permitiéndonos ver los valores obtenidos de aplicar el algoritmo DTW a cada una de las firmas 3D, lo que nos permite analizar las diferencias que puedan existir entre una firma y otra, realizadas por la misma persona, además de decidir el umbral máximo para determinar la aceptación o rechazo de la firma.

En la implementación del algoritmo DTW se ha utilizado la distancia Euclídea, para el cálculo de la matriz de distancias locales, debido a que es la más utilizada. Por otra parte, también se podría utilizar otras distancias como la distancia Manhattan o la mínima distancia, siendo necesario realizar cambios en el código de la aplicación.

Cuando se implemente el algoritmo DTW a cada una de las firmas, todas las aplicaciones de este algoritmo se deberán realizar siempre utilizando la misma distancia, ya sea Euclídea, Manhattan o mínima distancia, con el fin de evitar inconvenientes en el posterior análisis de los resultados.

Para disponer de una base de datos más sólida se han recogido cinco firmas por usuario, para de esa forma salvar las diferencias que puedan existir entre diferentes firmas de una misma persona y así disminuir los posibles falsos rechazos.

Todos estos resultados serán de utilidad a la hora de determinar la aceptación o rechazo de la firma. Por lo que, una vez determinado el umbral de decisión, se compara el valor de este con los resultados obtenidos del DTW, de encontrarse por debajo del umbral, la firma será aceptada, visualizándose un mensaje por la ventana de comandos indicando la correcta verificación y aceptación del usuario.

En caso de ser la firma mayor al umbral, esta será rechazada, indicando esta decisión a través de la ventana de comandos y con esto tenemos implementado todas las fases necesarias para el desarrollo del sistema de reconocimiento.

4.3.5 Estructura del programa

El programa se encuentra organizado en 4 funciones, cumpliendo cada una un proceso específico en el desarrollo de verificación de la firma. En la tablas 7 y 8 se han detallado las funciones y variables empleadas para el desarrollo del programa, contando con sus respectivas descripciones del desempeño que realizan cada una en el software.

		Descripción
Funciones	***Main()***	Ejecuta las funciones en un orden establecido para posteriormente mostrar por la ventana de comandos el resultado obtenido de aplicar el algoritmo.
	Numero_valores()	Accede a los ficheros patrón y captura y determina el número de datos que contiene cada uno, devuelve al programa principal dicho valor.
	Valor_DTW(double valor_normalizar)	Llamada desde la función principal con un argumento, siendo este el coste normalizado, necesario para obtener el DTW normalizado en longitud. Esta función aplica el algoritmo DTW a cada una de las características y retorna el resultado.
	Lectura_caracteristicas(int contador)	Llamada desde la función principal con una argumento, siendo este utilizado para establecer las dimensiones de la matriz que almacenarán los valores de cada una de las características. Lee los ficheros de datos y almacena los mismos en el array correspondiente.

Tabla 7. Funciones del programa de implementación del algoritmo DTW

		Descripción
Variables	***giroscopio_x_bs***	Variables utilizadas para almacenar los valores obtenidos de la base de datos de cada una de las características, así como también los valores capturados en el momento de la firma para la verificación de la misma.
	giroscopio_y_bs	
	giroscopio_z_bs	
	acelerometro_x_bs	
	acelerometro_y_bs	
	acelerometro_z_bs	
	giroscopio_x	
	giroscopio_y	
	giroscopio_z	
	acelerometro_x	
	acelerometro_y	
	acelerometro_z	
	ruta_1	Establece la ruta de lectura de los ficheros patrones.
	dat	Completa la ruta del fichero, por ejemplo, si existen 5 ficheros patrón, esta variable se incrementa en una unidad hasta alcanzar el valor 5 que sumado a ruta_1 en formato string completa la ruta de acceso: patron_**1.**txt
	file	Puntero a fichero, usado para acceder al mismo.
	linea	Almacena el valor leído del fichero para posteriormente adjudicarlo a el array de características correspondiente.
	dimension_matriz	Determina la dimensión de cada matriz.
	useSlopeConstraint	Variables utilizadas para establecer los parámetros necesarios para el correcto funcionamiento del algoritmo.
	slopeConstraintDiagonal	
	slopeConstraintAside	
	sakoeChibaMaxShift	
	useSakoeChibaMaxShift	
	resultado	Almacena la media correspondiente a la suma de los valores obtenidos del DTW, uno para cada característica.
	cost_1	Almacenan el valor obtenido de aplicar el algoritmo DTW a cada una de las características.
	cost_2	
	cost_3	
	cost_4	
	cost_5	
	longitud_base	Almacena el número de datos del fichero patrón
	longitud_cp	Almacena el número de datos del fichero captura
	file_cp	Puntero a fichero, usado para acceder al que contiene los datos de captura.
	valor	Utilizado en la función principal para almacenar el resultado devuelto de aplicar el algoritmo.
	resultado_DTW	Almacena el resultado final de aplicar el algoritmo DTW, utilizada para mostrar el resultado por la ventana de comandos.
	longitud_x	Almacenan la longitud de datos que contiene cada característica, son utilizadas para calcular el coste normalizado.
	longitud_y	

Tabla 8. Variables del programa de implementación del algoritmo DTW

Capítulo 5

Análisis de los resultados

Una vez que disponemos de todas las herramientas necesarias, continuamos con el análisis de las diferentes características con las que se cuenta en el proceso de reconocimiento de una firma 3D. Como se ha indicado anteriormente, en el reconocimiento biométrico, la firma depende del comportamiento de las personas por lo que un mismo usuario no es capaz de realizar su firma exactamente igual.

Debido a esto, para obtener unos datos patrón con los que comparar posteriormente la firma capturada, se han recogido 5 firmas de cada usuario, lo que nos permite observar las diferencias que existen entre ellas y que posteriormente aplicando el algoritmo DTW, nos permitirá establecer un umbral que decidirá el rechazo o aceptación en el proceso de verificación, disminuyendo de esta forma los posibles falsos rechazos y falsas aceptaciones.

Se ha determinado la necesidad de cinco firmas patrones disponibles en la base de datos con el objetivo de obtener las posibles variaciones a la hora de realizar una firma el mismo usuario y que el sistema sea capaz de verificar la autenticidad de la firma sin rechazarla.

Además también se ha determinado la obtención de 20 firmas falsas por usuario, con conocimiento previo de la firma por parte del usuario que realiza la falsificación, esto con el objetivo de que el sistema sea capaz de rechazar firmas falsas.

Por lo que, para la determinación del umbral de decisión se hacen uso de 5 firmas patrones más una capturada, además también se cuenta con 20 firmas falsas por usuario, esto nos permite posteriormente determinar los parámetros FAR y FRR y obtener la tasa de equierror (ERR) siendo este el punto de corte de las curvas correspondientes a los parámetros FAR y FRR y en donde se obtiene un equilibrio entre ambos parámetros.

Por último se realiza un análisis de cómo un menor número de características puede afectar a la fiabilidad del sistema de reconocimiento.

5.1 Análisis de las características

Contamos con seis características, tres correspondientes a la aceleración lineal y tres a la velocidad angular respecto a cada eje X, Y y Z. Para establecer el poder discriminante de las características se utiliza el Índice de Fisher, además de ver las diferencias de las características de distintas firmas de forma gráfica.

5.1.1 Comparación gráfica de características

Como un primer estudio, se mostrará de forma gráfica como las características toman valores distintos entre firmas de usuarios diferentes, obteniendo las respectivas curvas para cada firma. En la figura 28 se observa la variación de la velocidad angular en el eje x respecto del tiempo, por lo que se puede apreciar el cambio del valor obtenido según se esté realizando la firma.

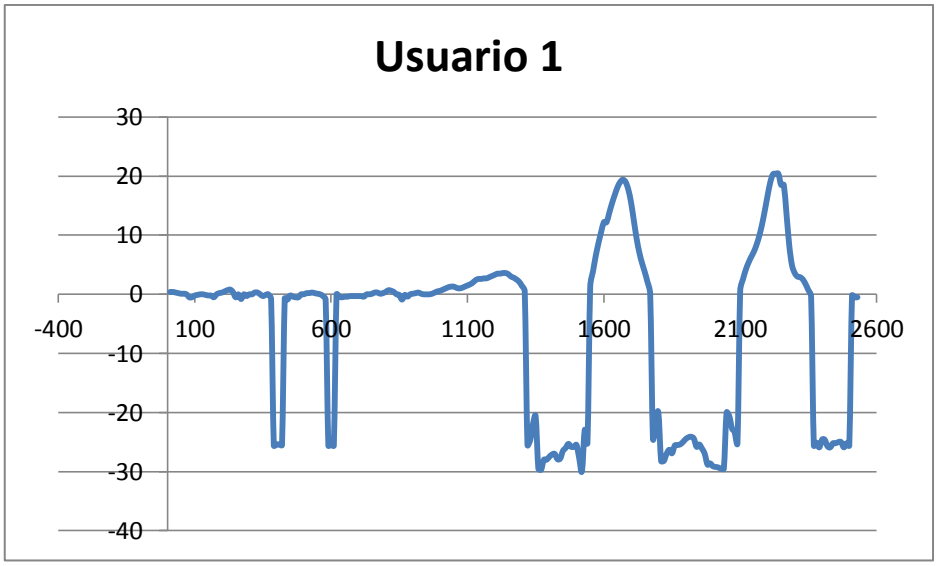

Figura 28. Variación de la velocidad angular en el eje x respecto al tiempo del usuario 1

En la figura 29 se muestra la firma de un segundo usuario y si la comparamos con la anterior, se aprecia una diferencia notable entre las mismas, en la figura 30 se observa esta comparación.

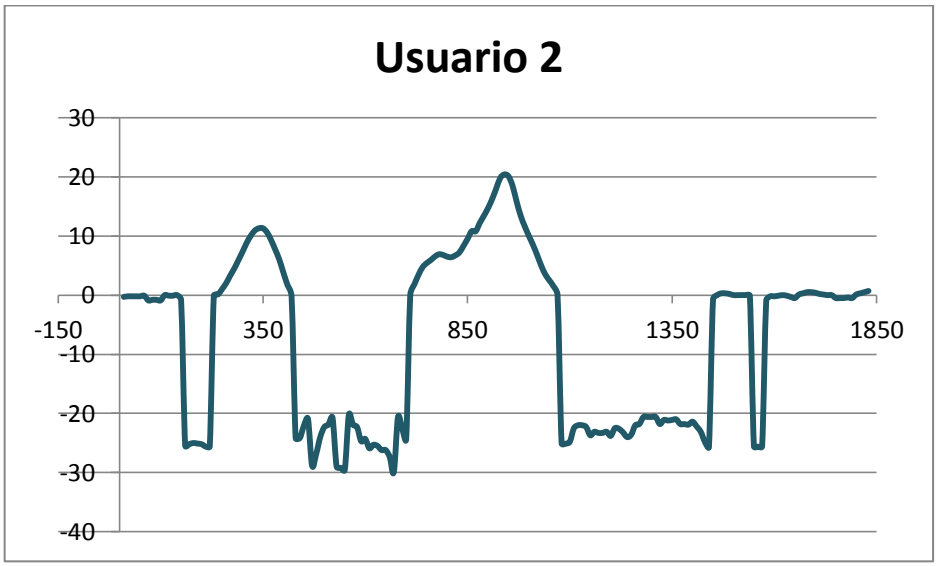

Figura 29. Variación de la velocidad angular en el eje x respecto al tiempo del usuario 2

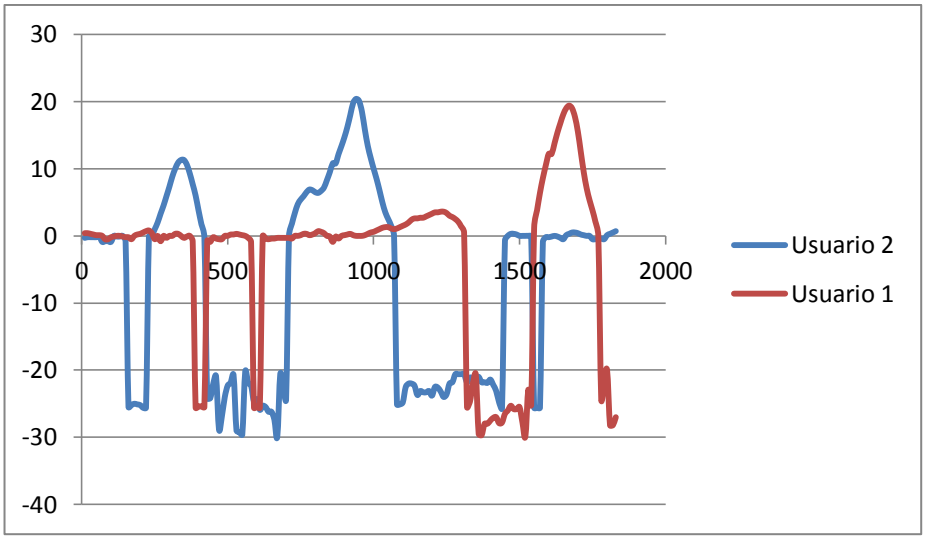

Figura 30. Comparación del parámetro velocidad angular en X de firmas de dos usuarios

Continuando con la comparación del resto de características también observamos diferencias notables entre los valores obtenidos de diferentes firmas, esto nos permite tener una primera idea del poder discriminante que tienen las características empleadas en el proceso de verificación de una firma 3D.

Las diferencias en los datos obtenidos de la velocidad angular en el eje Y son notables, los mismos se pueden apreciar en la figura 31. En la figura 32, en cambio se observa menos variación, pero no existen coincidencias entre las dos firmas lo que se traducirá posteriormente en un valor más alto de DTW dado por la aplicación del algoritmo.

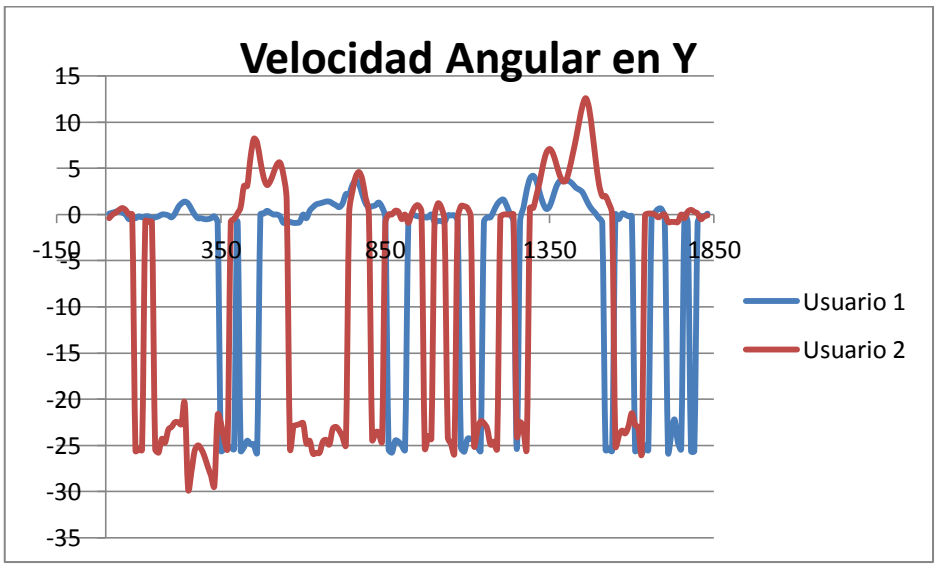

Figura 31. Comparación del parámetro velocidad angular en Y de firmas de dos usuarios

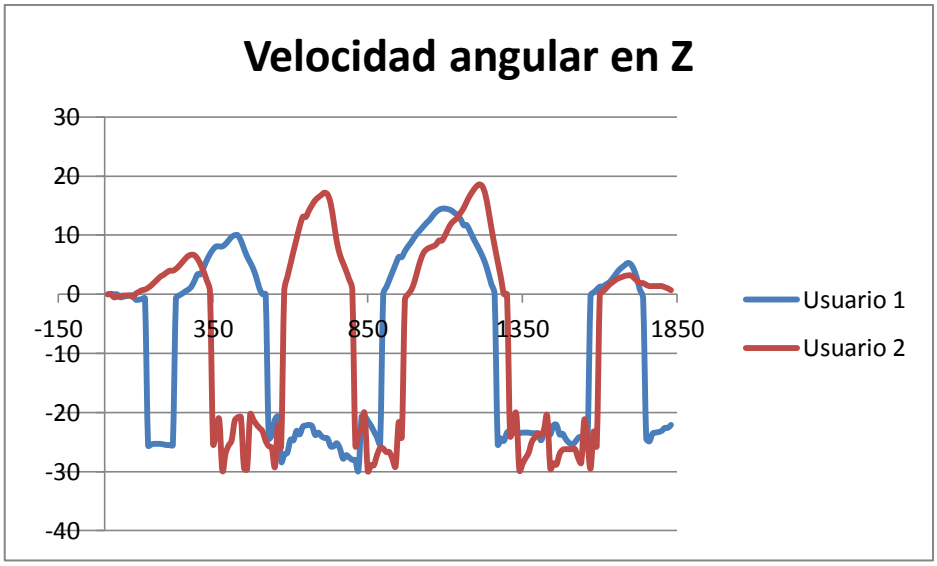

Figura 32. Comparación del parámetro velocidad angular en Z de firmas de dos usuarios

Resultados similares se observan en las figuras 33 y 34, en donde se comparan la aceleración lineal de los ejes Y y Z respectivamente.

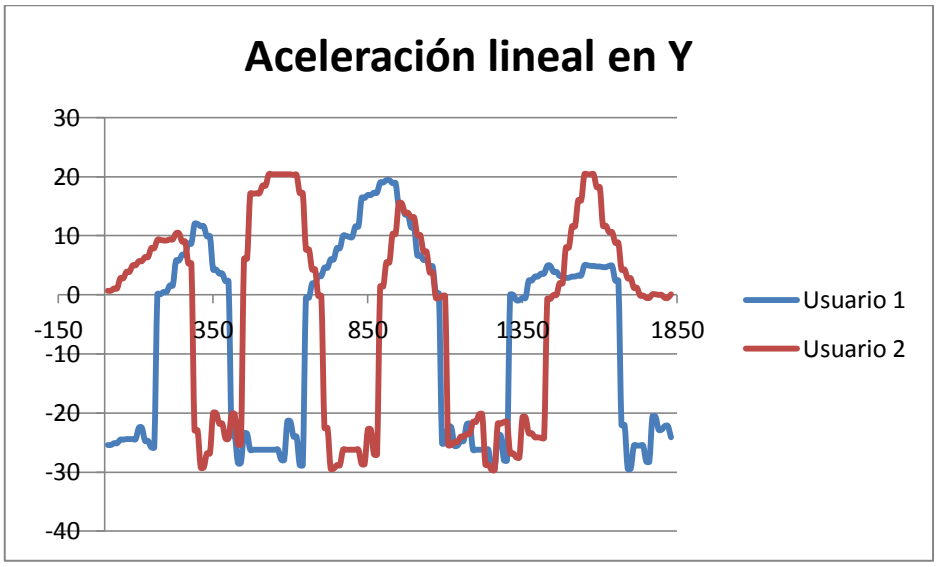

Figura 33. Comparación del parámetro aceleración lineal en Y de firmas de dos usuarios

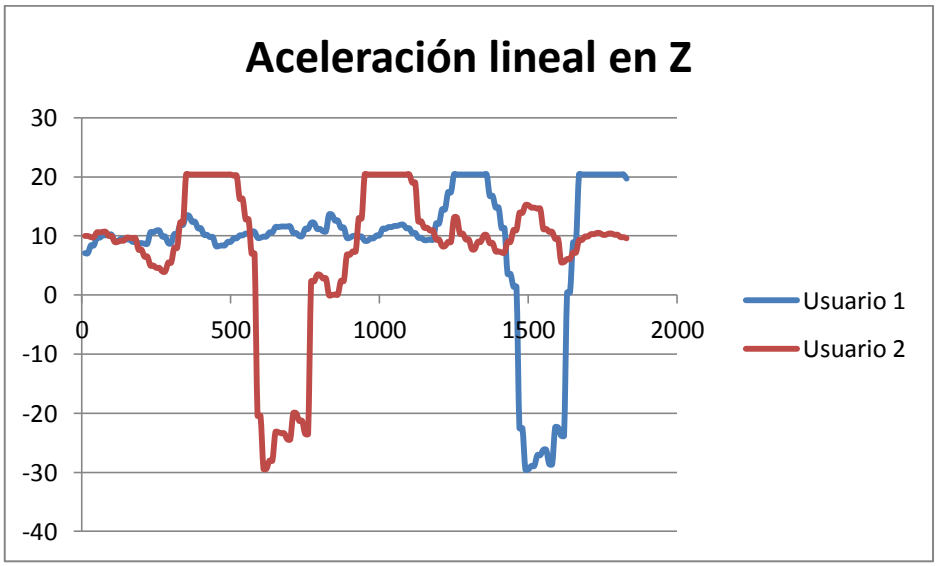

Figura 34. Comparación del parámetro aceleración lineal en Z de firmas de dos usuarios

Una vez realizada la comparación entre firmas de distintos usuarios, procedemos a realizar un segundo análisis correspondiente a observar las similitudes entre una misma firma ejecutada varias veces por un usuario.

Este proceso se efectúa para comprobar que existe similitud entre firmas de una misma persona, ya que de no ser así la característica no nos aportaría datos importantes siendo incluso un inconveniente en la fase de decisión, pudiendo llegar a incrementar la tasa de error.

Como se ha indicado anteriormente se han recogido cinco firmas por usuario, siendo estas utilizadas como firmas patrones, como una primera demostración, a continuación se aportará la comparación de dos firmas de un mismo usuario.

En la figura 35 se observa como los datos obtenidos del sensor de la velocidad angular x de dos firmas de un mismo usuario siguen un mismo desarrollo a lo largo del tiempo, salvo pequeñas diferencias, siendo las mismas las que serán establecidas por el algoritmo DTW, que al ser mínimas, como se verá en el apartado *5.3 Determinación del umbral de decisión,* los resultados obtenidos por la aplicación del algoritmo serán pequeñas en comparación con firmas de distintos usuarios.

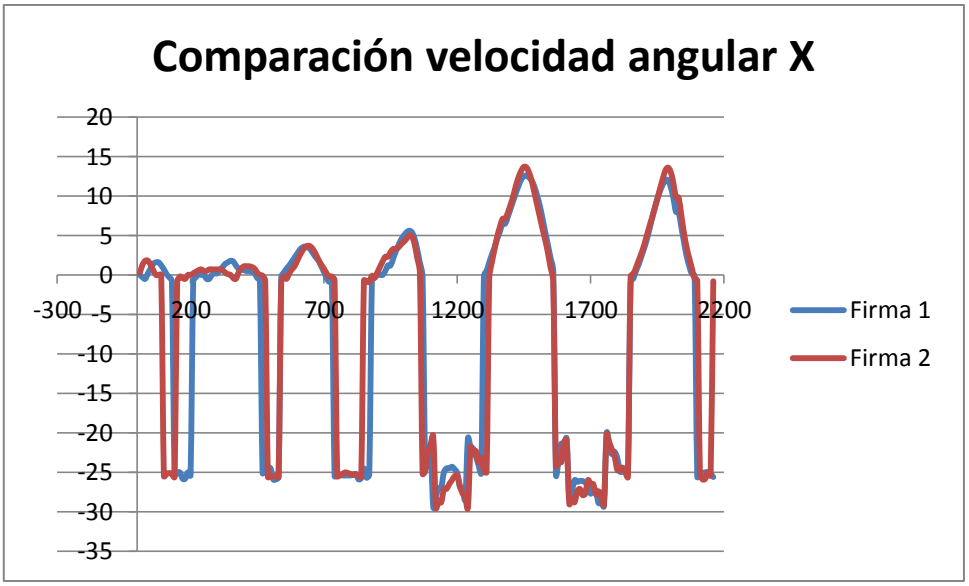

Figura 35. Comparación de la velocidad angular x de dos firmas de un mismo usuario

En cambio en la figura 36 se observa que las diferencias entre las firmas de un mismo usuario son sensiblemente mayores, pero también se aprecia ciertos períodos de tiempo en los que los valores de la velocidad angular en Y de ambas firmas son idénticos, esta similitud es lo que permite reducir el valor DTW que posteriormente se obtendrá con la aplicación del algoritmo, a diferencia de que si se comparan firmas de distintos usuarios, en donde no existirán tramos de patrones idénticos como en este caso, pudiendo ser diferenciadas.

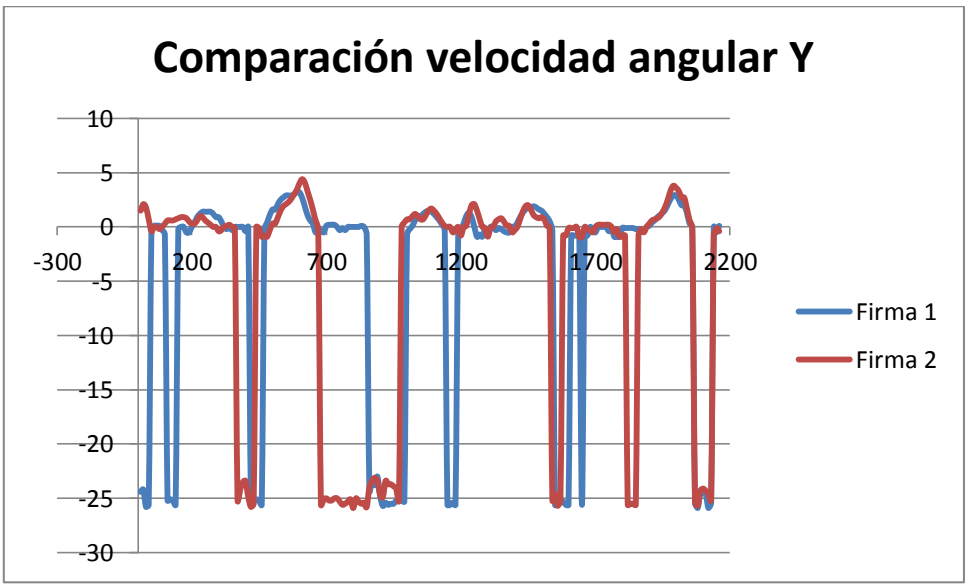

Figura 36. Comparación de la velocidad angular *Y* de dos firmas de un mismo usuario

En la figura 37 se representan los valores de la velocidad angular en z respecto del tiempo, se observa la misma tendencia que en la figura 34, existiendo pequeñas diferencias, pero que aportará relevancia al momento de la verificación.

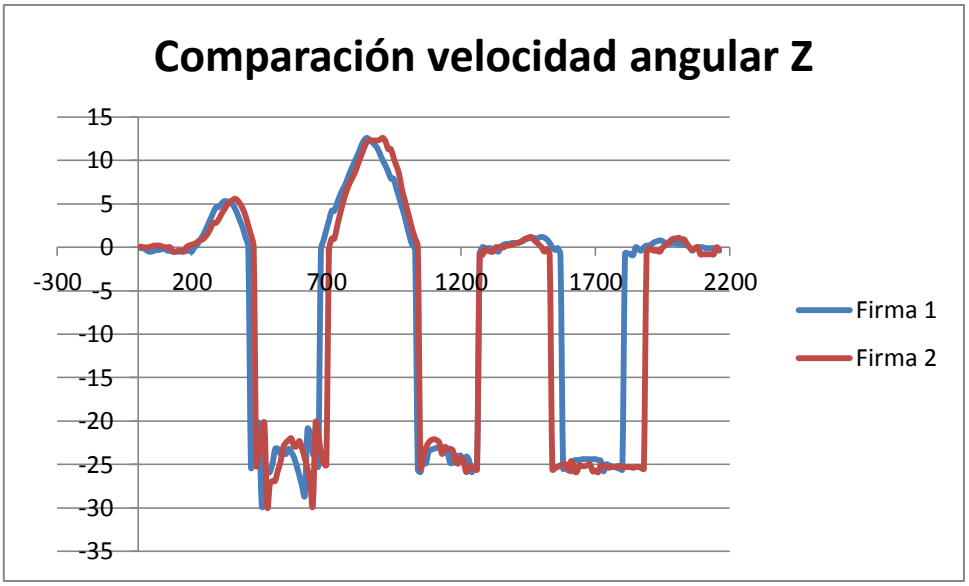

Figura 37. Comparación de la velocidad angular z de dos firmas de un mismo usuario

En la figura 38 se comparan los valores de la aceleración lineal en z, donde se observan trazos muy similares. En el siguiente apartado se confirmará el poder discriminante de las características mediante el índice Fisher.

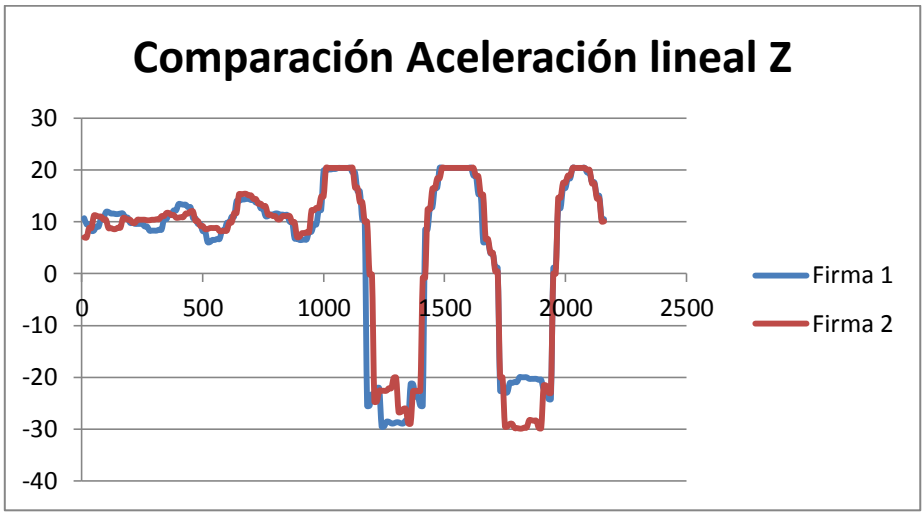

Figura 38. Comparación de la aceleración lineal Z y de dos firmas de un mismo usuario

5.1.2 El Índice Fisher

El Índice Fisher se utiliza para medir el poder discriminante de las características. Se expresa la idea de que la variabilidad de los valores de una característica en las firmas genuinas de un mismo usuario debe ser pequeña, mientras que la variabilidad entre firmas originales de distintos usuarios debe ser grande [17].

Con el fin de calcular esto, la variabilidad de una característica de la firma de un mismo usuario se ha medido como la variabilidad media de la característica para cada usuario. La variabilidad dentro del conjunto de firmas genuinas de un mismo usuario debe ser pequeña, lo que significa que esta característica es estable en cada una de las muestras de firmas originales de un mismo usuario. Sin embargo, la variabilidad entre usuarios, se calcula como la variabilidad

de la media de las características entre todos los usuarios, un valor grande indica que esta característica es lo suficientemente distintiva para cada usuario. Cuanto mayor sea el índice Fisher, más alto es el poder de discriminación de la característica estudiada [17], pudiendo obtener como resultados valores comprendidos entre 0 y 1.

El Índice Fisher se calcula aplicando la ecuación [5].

$$FR = \frac{\frac{1}{N} \cdot \sum_{j=1}^{N} (m_j - \overline{m})^2}{\frac{1}{N \cdot p} \cdot \sum_{j=1}^{N} \sum_{i=1}^{p} (x_{ji} - m_j)^2} \quad [5]$$

De donde:

$$\overline{m} = \frac{1}{N} \sum_{j=1}^{N} m_j \quad [6]$$

$$m_j = \frac{1}{p} \sum_{i=1}^{p} x_{ji} \quad [7]$$

Donde $\overline{m}$ es la media del valor de la característica considerando todos los usuarios, m_j es la media de la característica "x" del usuario "j", x_{ji} es la muestra "i" de la característica "x" del usuario "j", N es el número de usuarios y p es el número de muestras por usuario.

Dado que la varianza se ve afectada por el rango de valores que puedan tomar las características, es necesario realizar una normalización de los datos. Por lo que para evitar este problema, todos los valores de las características se normalizan entre valores de 0 y 1 [17].

5.1.3 Cálculo del Índice Fisher

Se procede a analizar cada una de las seis características para establecer el poder discriminativo de las mismas, confirmando de esta forma la importancia a la hora de verificar la firma. De tener una alta relevancia, dentro del proceso de decisión, ayudará a establecer su aceptación o rechazo, mientras que de tener poca relevancia podría elevar la tasa de error en la fase de decisión, de ahí la importancia de establecer su poder discriminante.

Para normalizar los datos empleamos la fórmula [8].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad [8]$$

Donde x' es el valor normalizado, x el valor a normalizar, min(x) el valor mínimo de la característica de entre las firmas de un mismo usuario y max(x) el valor máximo de la característica de entre las firmas de un mismo usuario.

Una vez normalizados los datos se ha empleado la fórmula [5] para el cálculo del Índice de Fisher expuesta en el apartado anterior, para su aplicación se ha utilizado una hoja de cálculo,

específicamente el programa Excel, ya que facilita el realizar esta operación, debido a la cantidad de datos a manejar y a la facilidad de la implementación de fórmulas de la que este programa dispone. Se obtuvieron los siguientes resultados:

	Características					
Índice de Fisher	**1**	**2**	**3**	**4**	**5**	**6**
Por unidad	0,8617	0,9212	0,8869	0,8976	0,7669	0,7998
En %	86,17	92,12	88,69	89,76	76,69	79,98

Tabla 9. Resultados del Índice de Fisher para cada característica.

De la tabla 9 se puede observar que todas las características han obtenido como resultado porcentajes mayores al 75% en poder de discriminación, por lo que se verifica la relevancia que las mismas tendrán en el proceso de verificación y se confirma de que ninguna de ellas supondrá un problema en la fase de decisión, es decir, que todas las características son capaces de diferenciar entre firmas de distintos usuarios y que no supondrán una influencia negativa en la tasa de error.

5.2 Base de datos

La base de datos que posteriormente será empleada para la determinación del umbral de decisión y comprobación del funcionamiento del sistema, ha sido construida utilizando el programa de adquisición de datos expuesto en el apartado *4.2 Desarrollo del software de recepción y almacenamiento de datos*.

La misma está compuesta por cinco usuarios, de los cuales se han capturado 5 firmas genuinas y 20 firmas falsas por persona. Para la captura de las firmas falsas los usuarios tenían conocimiento previo de la firma a imitar, de este modo se evaluará las posibilidades de falsas aceptaciones.

5.3 Determinación del umbral de decisión

Una vez que se ha comprobado la relevancia de los datos en el proceso de verificación, tenemos que establecer el umbral que determinará si una firma es genuina o falsa, para ello se obtendrán los valores de aplicar el DTW a las firmas existentes en la base de datos. Cabe destacar que cuanto más coincidan las firmas, el resultado obtenido del DTW será más bajo produciéndose el resultado contrario en el caso de firmas muy distintas, arrojando valores elevados. Los valores del DTW están comprendidos entre 0 y 100.

Como una primera etapa se determinó el valor del umbral de decisión considerando toda la base de datos, estableciendo de esta forma un único valor de umbral que determine la autenticidad de cualquiera de las firmas disponibles en la base de datos.

En una segunda etapa se evaluó cada firma para obtener un valor de umbral de decisión para cada una de ellas, es decir, el valor del umbral de decisión no tiene porque ser el mismo para todas las firmas.

5.3.1 Umbral de decisión conjunto

Para determinar el valor del umbral de decisión conjunto se considera todas las firmas disponibles en la base de datos. En este caso se obtiene un único valor del umbral que determinará la aceptación o rechazo de una firma.

Para el cálculo de la tasa de falsas aceptaciones y la tasa de los falsos rechazos se ha realizado lo siguiente:

- Se ha empezado con un valor inicial del umbral igual a cinco, siendo incrementado el mismo cinco unidades hasta un valor máximo de 100.
- Para cada valor del umbral, se ha comparado el valor DTW de cada firma falsa, si el mismo era inferior al umbral se contabilizaba como una firma falsa aceptada.
- Para cada valor del umbral, se ha comparado el valor DTW de cada firma genuina, si el mismo era superior al umbral se contabilizaba como una firma genuina rechazada.
- Para obtener la tasa de falsas aceptaciones (FAR) se ha divido el número de firmas falsas aceptadas entre el número total de firmas falsas.
- Por último, para obtener la tasa de falsos rechazos (FRR) se ha dividido el número de firmas genuinas rechazadas entre el número total de firmas genuinas.

En la tabla 10 se exponen los resultados finales de analizar todas las firmas disponibles en la base de datos y en la figura 39 se observa la gráfica que representa los valores de la tabla anterior.

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	25	0	0,83
10	0	24	0	0,80
15	0	19	0	0,63
20	0	16	0	0,53
25	0	8	0	0,27
30	6	1	0,06	0,03
35	25	0	0,25	0
40	37	0	0,37	0
45	48	0	0,48	0
50	59	0	0,59	0
55	61	0	0,61	0
60	74	0	0,74	0
65	86	0	0,86	0
70	94	0	0,94	0
75	99	0	0,99	0
80	100	0	1	0
85	100	0	1	0
90	100	0	1	0
95	100	0	1	0
100	100	0	1	0

Tabla 10. Resultados del análisis de la firmas que componen la base de datos

Como puede apreciarse en la figura 39 las curvas se cortan, las áreas azul y roja indican la posibilidad de que se produzca un error entre aceptación y rechazo respectivamente, aunque el área azul sea ligeramente mayor al área roja la tasa de error es muy baja, siendo esta de 0.046 que representado en porcentaje tendremos una tasa de equierror (ERR) del 4.6%, por lo

tanto se elige como umbral de decisión el punto de corte de las curvas siendo este de 29.5, este valor se lo establece en el programa *Aplicación del algoritmo DTW y fase de decisión.*

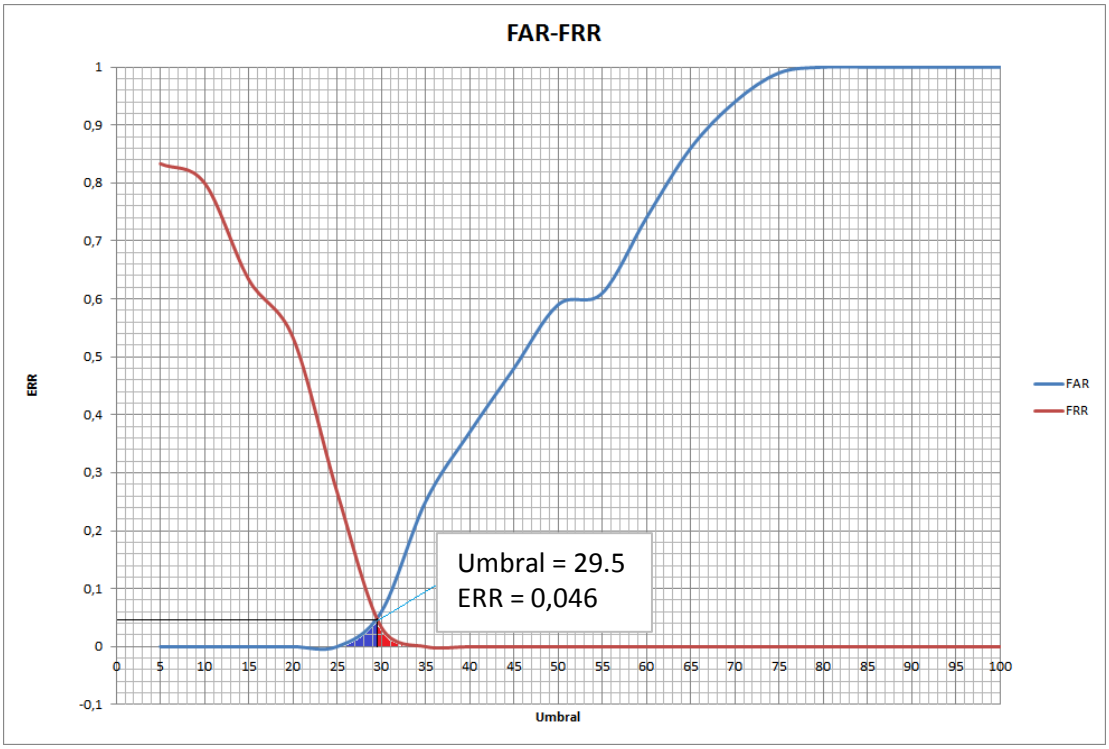

Figura 39. FAR y FRR de los cinco usuarios

5.3.2 Umbral de decisión individual

Una segunda opción sería determinar un valor del umbral de decisión para cada usuario disponible en la base de datos, con el objetivo de conseguir una disminución de la tasa de equierror y por ende un menor número de falsas aceptaciones y falsos rechazos. A continuación se exponen los resultados del valor del umbral de decisión para cada firma.

En la tabla 11 se muestran los resultados obtenidos por el algoritmo DTW, en donde la firma cuyo valor es cero corresponde a la firma capturada utilizada para la comparación con las demás firmas.

También se puede apreciar como el resultado del DTW de comparar la firma 1 con las siguientes cinco firmas es bajo, esto debido a que la primera firma corresponde a una firma capturada pero no almacenada en la base de datos y las cinco siguientes firmas son las que se encuentran disponibles en la base de datos y utilizadas como firmas patrones y que corresponden a un mismo usuario, lo que nos permite comprobar el correcto funcionamiento del algoritmo.

Firma 1	Firma 2	Firma 3	Firma 4	Firma 5
0	15,71	12,81	12,53	10,42
Firma 6	**Firma 7**	**Firma 8**	**Firma 9**	**Firma 10**
8,01	45,22	46,68	45,89	41,79
Firma 11	**Firma 12**	**Firma 13**	**Firma 14**	**Firma 15**
31,21	43,26	41,5	30,14	28,82
Firma 16	**Firma 17**	**Firma 18**	**Firma 19**	**Firma 20**
36,17	33,79	25,22	30,76	41,6
Firma 21	**Firma 22**	**Firma 23**	**Firma 24**	**Firma 25**
39,18	38,83	51,93	30,4	34,55
Firma 26				
33,96				

Tabla 11. Resultados del DTW usuario 1

Desde la firma 7 hasta la firma 26 se observan valores altos de DTW, siendo estos resultados esperados, debido a que dichas firmas no corresponden al usuario de la firma patrón utilizada para la comparación. Las mismas son las correspondientes a las firmas falsas realizadas por otro usuario.

Utilizando estos primeros resultados se obtendrán las gráficas correspondientes a las falsas aceptaciones (FAR) y los falsos rechazos (FRR) que nos permitirán establecer el umbral de decisión.

En la figura 40 se muestra la representación de las falsas aceptaciones y los falsos rechazos. En este caso se observa como la curva FRR empieza en valores altos hasta valores bajos de ERR, lo mismo pasa con la curva FAR, en donde empieza con valores bajos hasta valores altos. Los valores bajos del umbral indican similitud y los valores altos grandes diferencias.

En este caso, como umbral podemos elegir cualquier valor que se encuentre entre la curva FRR y la curva FAR de la figura 40. Como umbral de decisión se establece una valor de 25, donde la tasa de equierror es muy cercana a cero.

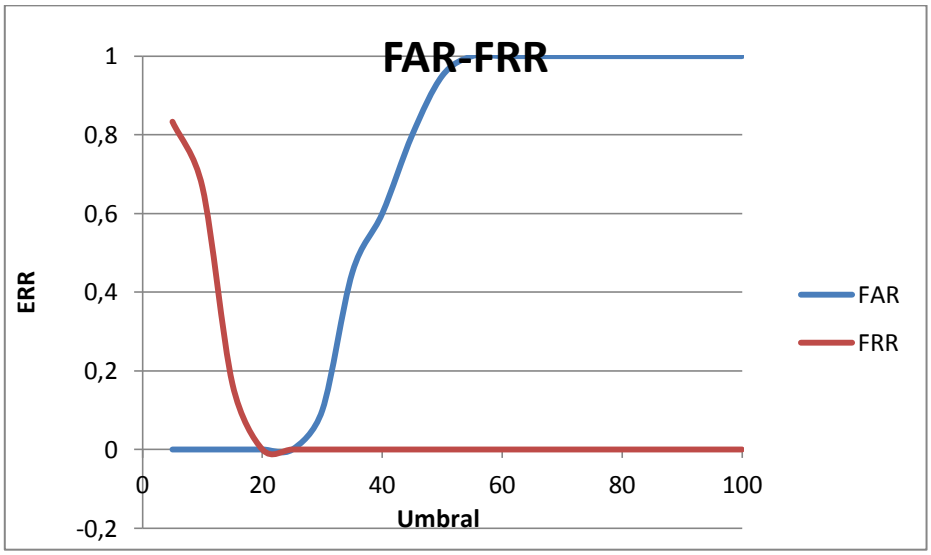

Figura 40. Curvas FAR y FRR para el usuario 1

En la tabla 12 se exponen los resultados obtenidos de comparar una firma capturada del usuario 2 con las firmas patrones, siendo desde la firma 1 hasta la firma 6 las rúbricas pertenecientes al mismo usuario y desde la firma 7 a la firma 26 las firmas falsificadas por otro usuario. Los resultados son similares a los del usuario 1, es decir, el valor obtenido del DTW de la comparación entre firmas de un mismo usuario son bajos en comparación a los obtenidos con las firmas falsas.

Firma 1	Firma 2	Firma 3	Firma 4	Firma 5
0	15,82	24,13	25,68	14,02
Firma 6	**Firma 7**	**Firma 8**	**Firma 9**	**Firma 10**
21,16	58,19	67,07	65,14	65,41
Firma 11	**Firma 12**	**Firma 13**	**Firma 14**	**Firma 15**
68,73	70,45	61,55	68,33	63,41
Firma 16	**Firma 17**	**Firma 18**	**Firma 19**	**Firma 20**
73,34	68,64	74,48	72,08	64,45
Firma 21	**Firma 22**	**Firma 23**	**Firma 24**	**Firma 25**
67,4	75,69	66,77	57,6	71,37
Firma 26				
64,64				

Tabla 12. Resultados del DTW usuario 2

En la figura 41 se muestran las curvas FRR y FAR obtenidas para el usuario 2, en este caso se observa como el posible umbral de decisión a elegir es más amplio que en el caso anterior, por lo que se puede establecer como umbral de decisión el valor 40 que se encuentra comprendido entre las curvas FRR y FAR y en donde la tasa de equierror es muy bajo.

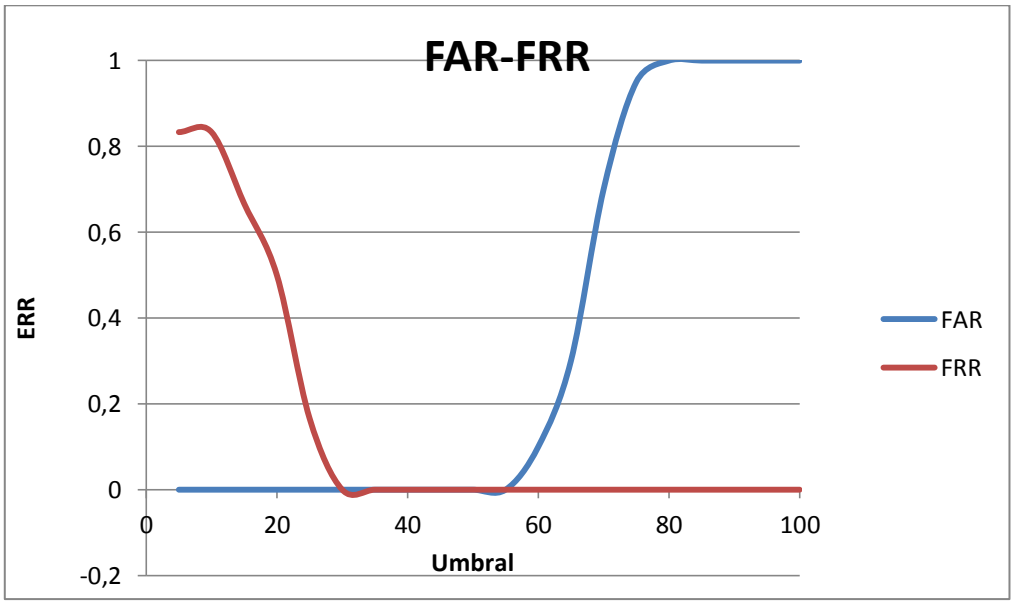

Figura 41. Curvas FAR y FRR para el usuario 2

Continuando con el análisis, en la tabla 13 se presentan los resultados del DTW para el usuario 3, en donde una vez más se aprecia un cambio en los posibles valores para el umbral de

decisión, debido a que el máximo valor de una firma genuina es de 28,11 y el mínimo valor para una firma falsa es de 54,41, encontrándose entre los mismos el valor del umbral a elegir.

Firma 1	Firma 2	Firma 3	Firma 4	Firma 5
0	30,2	25,69	28,04	28,11
Firma 6	**Firma 7**	**Firma 8**	**Firma 9**	**Firma 10**
24,54	62,31	58,57	55,83	58,62
Firma 11	**Firma 12**	**Firma 13**	**Firma 14**	**Firma 15**
59,2	61,06	59,25	60,96	60,54
Firma 16	**Firma 17**	**Firma 18**	**Firma 19**	**Firma 20**
64,46	64,64	58,29	61,82	63,84
Firma 21	**Firma 22**	**Firma 23**	**Firma 24**	**Firma 25**
59,31	59,19	55,58	54,41	58,03
Firma 26				
56,44				

Tabla 13. Resultados del DTW usuario 3

En la figura 42 perteneciente a los resultados de FAR y FRR para el usuario tres se observa que el umbral de decisión en este caso puede ser el mismo que el determinado para el usuario dos, es decir, se puede establecer 40 como valor del umbral ya que también se obtendría una tasa de equierror muy baja.

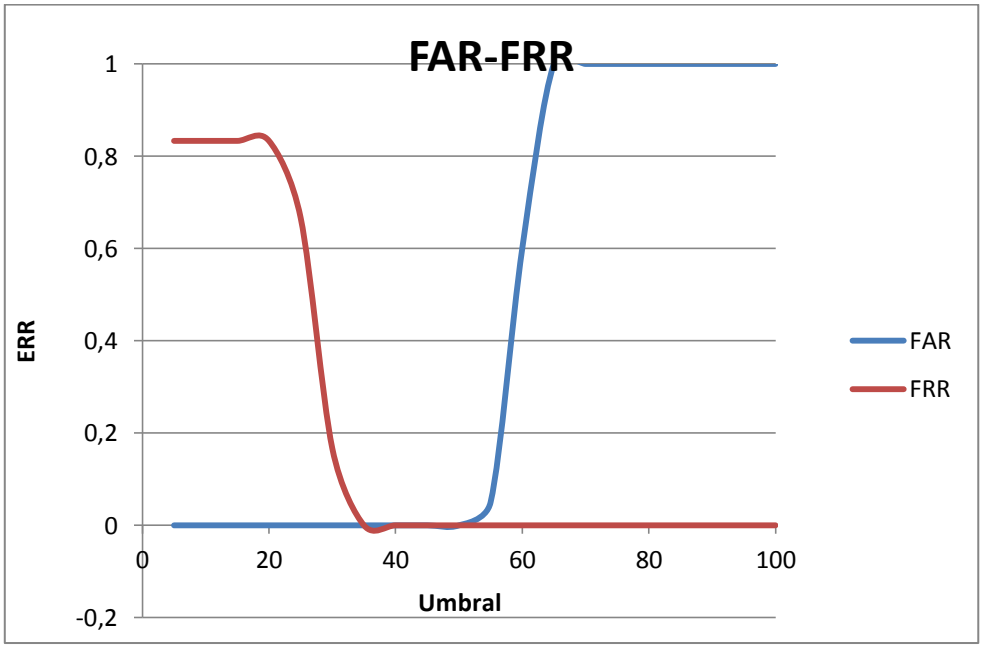

Figura 42. Curvas FAR y FRR para el usuario 3

Como se ha visto hasta ahora con los resultados de los tres usuarios analizados, el posible valor del umbral siempre varía entre unas y otras firmas de distintos usuarios. En las tablas 14 y 15 se muestran los resultados correspondientes al usuario 4 y usuario 5 respectivamente, siendo los valores obtenidos los esperados.

En relación a estos dos últimos usuarios se aprecia una diferencia en el posible valor del umbral, ya que se pasa de tener un rango de posibles valores a tener solo uno, siendo este el punto en donde se cortan las curvas.

Así para el usuario 4 tenemos un valor de umbral de 30, cuyas curvas FAR y FRR se encuentran representadas en la figura 43. Mientras que para el usuario 5 el umbral de decisión sería de 25 y sus curvas FAR y FRR se encuentran representadas en la figura 44.

Firma 1	Firma 2	Firma 3	Firma 4	Firma 5
0	26,38	25,5	20,5	23,11
Firma 6	**Firma 7**	**Firma 8**	**Firma 9**	**Firma 10**
27,53	44,93	35,76	38,96	40,32
Firma 11	**Firma 12**	**Firma 13**	**Firma 14**	**Firma 15**
34,06	36,76	48,01	40,76	33,95
Firma 16	**Firma 17**	**Firma 18**	**Firma 19**	**Firma 20**
32,33	31,25	34,94	28,8	34,67
Firma 21	**Firma 22**	**Firma 23**	**Firma 24**	**Firma 25**
31,13	33,21	29,77	30,24	37,49
Firma 26				
40,18				

Tabla 14. Resultados del DTW usuario 4

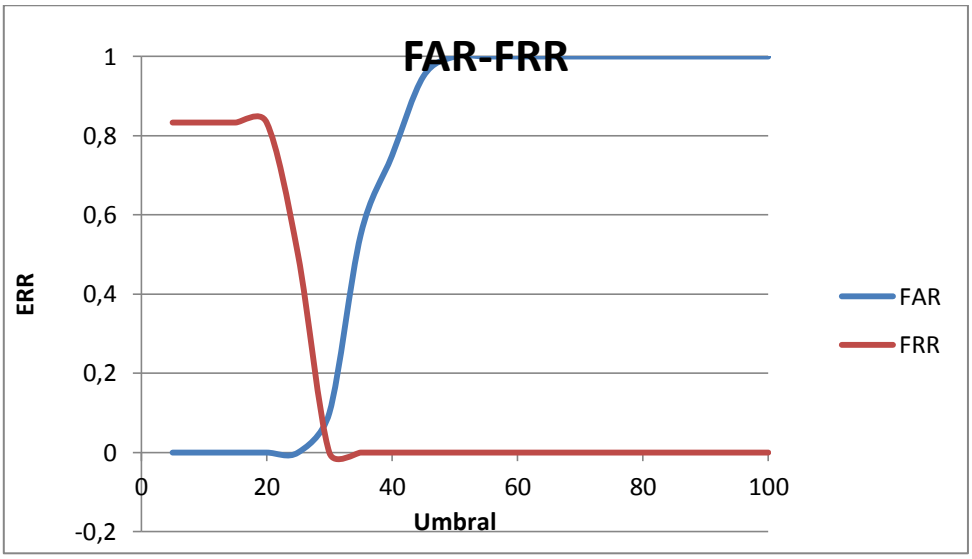

Figura 43. Curvas FAR y FRR para el usuario 4

Firma 1	Firma 2	Firma 3	Firma 4	Firma 5
0	23,7	20,25	21,05	14
Firma 6	**Firma 7**	**Firma 8**	**Firma 9**	**Firma 10**
19,15	31,09	41,59	31,12	37,88
Firma 11	**Firma 12**	**Firma 13**	**Firma 14**	**Firma 15**
49,37	37,1	46,96	45,14	48,63
Firma 16	**Firma 17**	**Firma 18**	**Firma 19**	**Firma 20**
46,92	37,95	48,28	44,87	45,63
Firma 21	**Firma 22**	**Firma 23**	**Firma 24**	**Firma 25**
44,42	33,58	35,32	29,21	39,81
Firma 26				
27,69				

Tabla 15. Resultados del DTW usuario 5

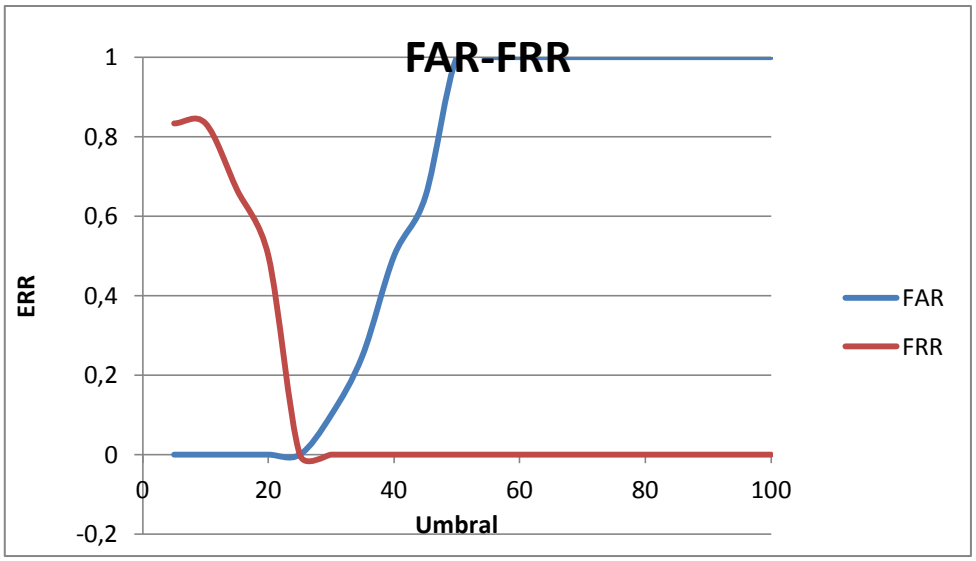

Figura 44. Curvas FAR y FRR para el usuario 5

De los resultados anteriores podemos concluir, como el umbral de decisión cambia para las firmas de cada usuario, por lo que en este caso se podría establecer un valor de umbral de decisión para cada usuario teniendo la ventaja de obtener una disminución de las tasas de falsas aceptaciones y falsos rechazos.

Se utilizó como umbral el valor obtenido en el apartado *5.3.1 umbral de decisión conjunto*, debido a que la tasa de equierror no presenta grandes diferencias entre la obtenida de forma conjunta y las obtenidas de forma individual. Además utilizar un umbral de decisión individual conllevaría un aumento de la complejidad del sistema con su respectivo aumento del coste computacional.

5.4 Determinación de la aceptación o rechazo de la firma

Como ya se ha indicado anteriormente los valores DTW menores al umbral de decisión darán como resultado la aceptación de la firma y valores superiores el rechazo de la misma. Para la obtención del valor final del DTW de la firma a verificar no se considera solamente la

comparación con una firma patrón, sino que se la compara con cinco firmas patrones, obteniendo así de cada una de ellas un valor de DTW, siendo el resultado final la media de las cinco aplicaciones del algoritmo, es decir:

$$DTW = \frac{DTW_1 + DTW_2 + DTW_3 + DTW_4 + DTW_5}{5} \quad [9]$$

Esta operación se realiza para evitar considerar un único valor que pueda dar como resultado el rechazo de una firma genuina, por lo que se la compara con otras cuatro firmas patrón en las que se asegura la consideración de las posibles variaciones que pueda tener la firma realizada por una misma persona.

5.5 Análisis del umbral de decisión con un menor número de características

Si la base de datos está formada por un gran número de usuarios, esto se traduce en una mayor cantidad de datos a procesar y esto a su vez en un coste computacional elevado a la hora de aplicar el algoritmo DTW.

Debido a esto se realiza un estudio de la tasa de error obtenida con un menor número de características, para determinar de esta forma si es conveniente la reducción de las mismas a favor de una mayor velocidad de respuesta del sistema e incluso si es posible su implementación en dispositivos que dispongan poca memoria o una capacidad de procesamiento inferior a la de un ordenador.

Por consiguiente un menor número de características conlleva a que el sistema de reconocimiento sea más rápido y use menos memoria, por otra parte, existe la posibilidad de que el empleo de un menor número de características conduzca a una pérdida en el poder de discriminación y por lo tanto menor será la precisión del sistema de reconocimiento resultante [18].

El análisis se realizará mediante un sistema de selección decreciente conocido como selección secuencial hacia atrás [19], es decir, se disminuirá una característica en cada caso, obteniéndose la tasa de error de utilizar ese determinado número de características.

Para establecer el orden en el que se eliminan las características se hace uso de los resultados obtenidos en el apartado *5.1.3 Cálculo del Índice Fisher,* en donde se ha obtenido el poder discriminante de cada una de las características, por lo tanto en cada iteración se elimina la característica cuyo poder de discriminación sea menor.

La primera característica que se elimina es la número cinco, correspondiente con la aceleración lineal en el eje z, manteniendo las otras cinco características (a_x, a_y, ω_x, ω_y, ω_z) y siguiendo el mismo proceso detallado en el apartado *5.3 Determinación del umbral de decisión.*

Los resultados obtenidos son los siguientes:

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	25	0	0,83
10	0	24	0	0,80
15	0	21	0	0,70
20	0	17	0	0,57
25	0	11	0	0,37
30	1	2	0,01	0,07
35	19	0	0,19	0
40	37	0	0,37	0
45	47	0	0,47	0
50	53	0	0,53	0
55	61	0	0,61	0
60	73	0	0,73	0
65	82	0	0,82	0
70	84	0	0,84	0
75	94	0	0,94	0
80	97	0	0,97	0
85	100	0	1,00	0
90	100	0	1,00	0
95	100	0	1,00	0
100	100	0	1,00	0

Tabla 16. Resultados considerando cinco características.

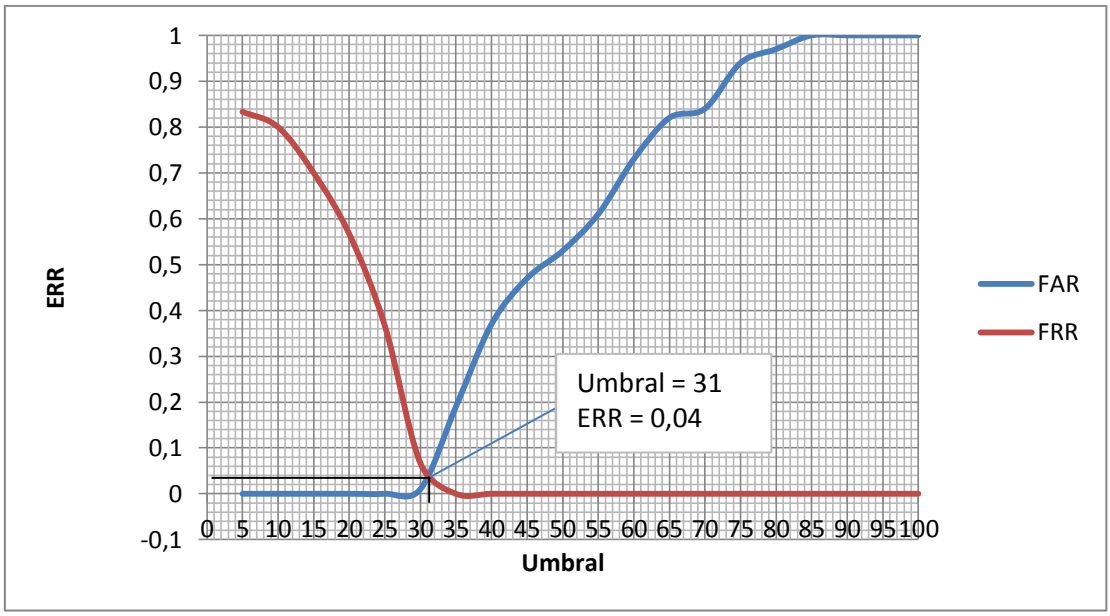

Figura 45. Curvas FAR y FRR con cinco características.

En este caso se obtiene un valor para el umbral de decisión mayor al determinado con todas las características, así mismo se observa una tasa de error ligeramente menor a la obtenida con el sistema completo en donde el ERR es de 0.046.

Un valor menor de la tasa de equierror supondría una mayor fiabilidad del sistema de reconocimiento, pero la diferencia entre el ERR del sistema completo y el que se analiza con 5 características es del 0.6%, lo que no supondría una gran mejora en la fiabilidad, pero sí confirmaría la validez del sistema si se utilizan sólo cinco características.

La siguiente característica que se elimina es la número seis correspondiente con la velocidad angular en el eje z, por lo que se considera sólo cuatro características (a_x, a_y, ω_x, ω_y) para la siguiente iteración.

Se obtienen los siguientes resultados:

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	25	0	0,83
10	0	22	0	0,73
15	0	20	0	0,67
20	0	17	0	0,57
25	0	11	0	0,37
30	3	3	0,03	0,10
35	11	0	0,11	0
40	33	0	0,33	0
45	46	0	0,46	0
50	53	0	0,53	0
55	61	0	0,61	0
60	71	0	0,71	0
65	79	0	0,79	0
70	86	0	0,86	0
75	93	0	0,93	0
80	98	0	0,98	0
85	100	0	1,00	0
90	100	0	1,00	0
95	100	0	1,00	0
100	100	0	1,00	0

Tabla 17. Resultados considerando cuatro características.

En la figura 45 se aprecia como el umbral de decisión aumenta al igual que la tasa de equierror en comparación con los dos casos anteriores.

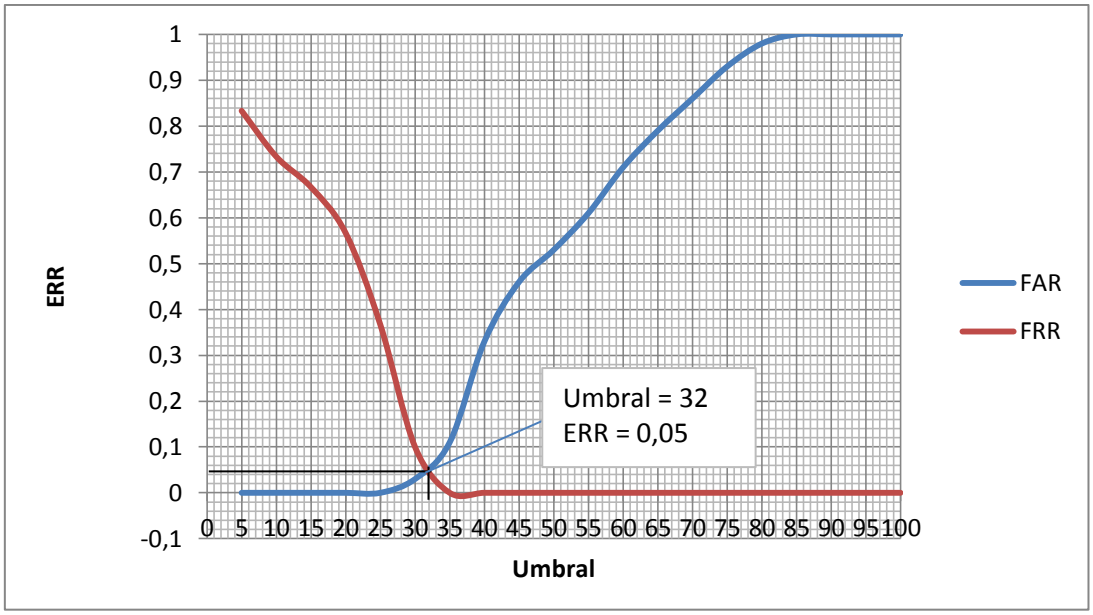

Figura 46. Curvas FAR y FRR con cuatro características.

La siguiente característica que se elimina es la número uno que corresponde a la aceleración lineal en el eje x, considerando en esta iteración sólo tres características (a_y, ω_x, ω_y).

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	25	0	0,83
10	0	22	0	0,73
15	0	19	0	0,63
20	0	18	0	0,60
25	1	12	0,01	0,40
30	8	5	0,08	0,17
35	21	2	0,21	0,07
40	34	0	0,34	0
45	42	0	0,42	0
50	58	0	0,58	0
55	68	0	0,68	0
60	76	0	0,76	0
65	80	0	0,8	0
70	80	0	0,8	0
75	80	0	0,8	0
80	83	0	0,83	0
85	88	0	0,88	0
90	93	0	0,93	0
95	99	0	0,99	0
100	100	0	1,00	0

Tabla 18. Resultados considerando tres características.

En la figura 47 se observa como el punto de corte entre las dos curvas corresponde a una tasa de equierror de 0.12, siendo este un valor sensiblemente alto en comparación con los casos anteriores en donde se obtenían tasas inferiores a 0.05. Por lo que si se emplea sólo tres características en el sistema de reconocimiento la fiabilidad del mismo se vería disminuida.

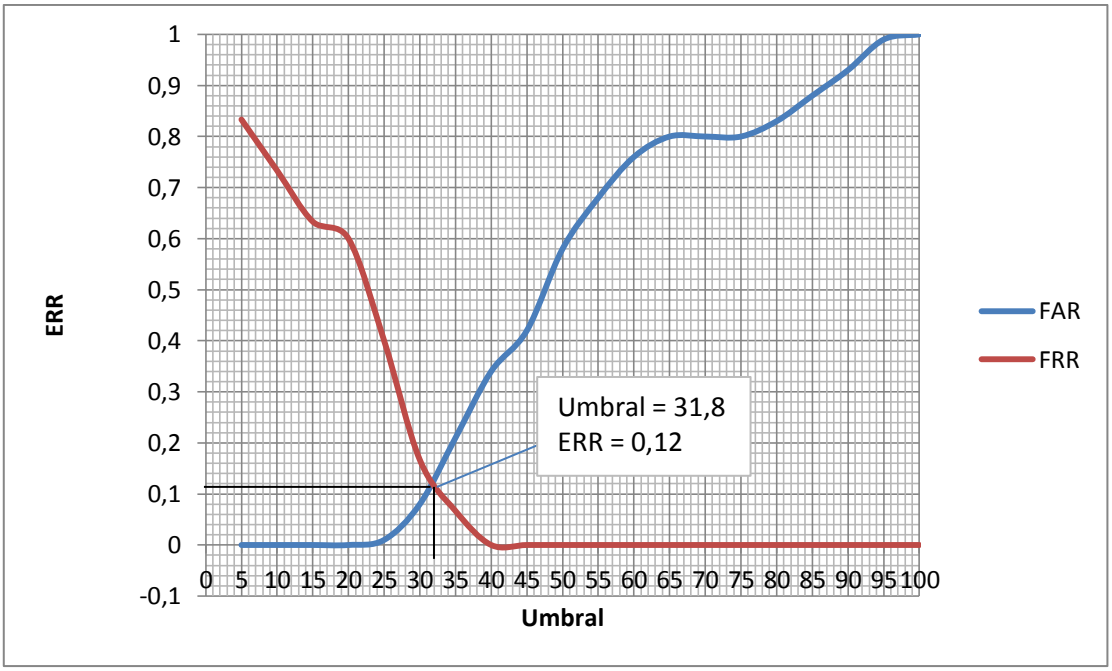

Figura 47. Curvas FAR y FRR con tres características.

En la siguiente iteración se elimina la característica número tres correspondiente con la velocidad angular en el eje Y, considerando en este caso en el sistema de reconocimiento sólo dos características (a_y, ω_x), obteniéndose los siguientes resultados:

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	25	0	0,83
10	0	21	0	0,70
15	0	19	0	0,63
20	0	16	0	0,53
25	1	12	0,01	0,40
30	12	4	0,12	0,13
35	16	0	0,16	0
40	27	0	0,27	0
45	35	0	0,35	0
50	50	0	0,5	0
55	54	0	0,54	0
60	63	0	0,63	0
65	70	0	0,7	0
70	77	0	0,77	0
75	80	0	0,8	0
80	80	0	0,8	0
85	81	0	0,81	0
90	82	0	0,82	0
95	86	0	0,86	0
100	100	0	1,00	0

Tabla 19. Resultados considerando dos características.

En este caso se observa un aumento considerable de la tasa de equierror, por lo que el uso de sólo dos características haría menos fiable al sistema de reconocimiento. En la figura 48 se obtiene que la tasa de equierror es de 0.126 y el umbral de 30.

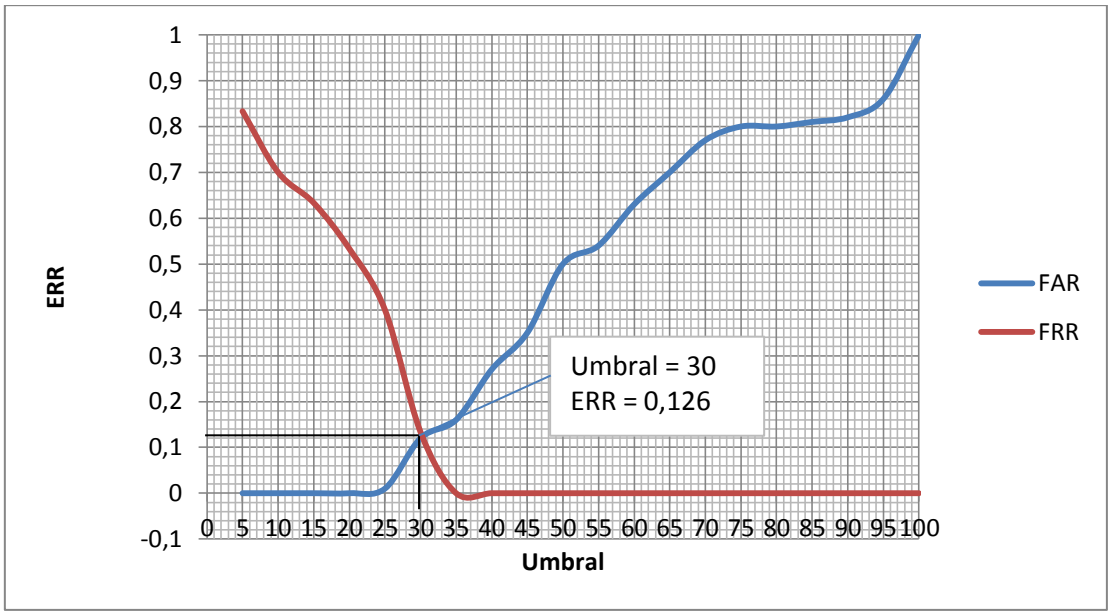

Figura 48. Curvas FAR y FRR con dos características.

Para la última iteración se elimina la característica número cuatro correspondiente con la aceleración lineal en Y, por lo que se considera una única característica siendo esta la velocidad angular en x, consiguiendo los siguientes resultados:

Umbral	Nº Firmas falsas aceptadas	Nº Firmas Genuinas Rechazadas	FAR	FRR
5	0	23	0	0,77
10	1	20	0,01	0,67
15	2	18	0,02	0,60
20	4	15	0,04	0,50
25	14	8	0,14	0,27
30	24	6	0,24	0,20
35	37	0	0,37	0
40	47	0	0,47	0
45	54	0	0,54	0
50	59	0	0,59	0
55	63	0	0,63	0
60	67	0	0,67	0
65	73	0	0,73	0
70	78	0	0,78	0
75	81	0	0,81	0
80	84	0	0,84	0
85	85	0	0,85	0
90	87	0	0,87	0
95	92	0	0,92	0
100	100	0	1,00	0

Tabla 20. Resultados considerando una característica.

En la figura 49 se observa como la tasa de equierror pasa a ser de 0.22, siendo totalmente descartada la posibilidad de que el sistema de reconocimiento utilice una única característica.

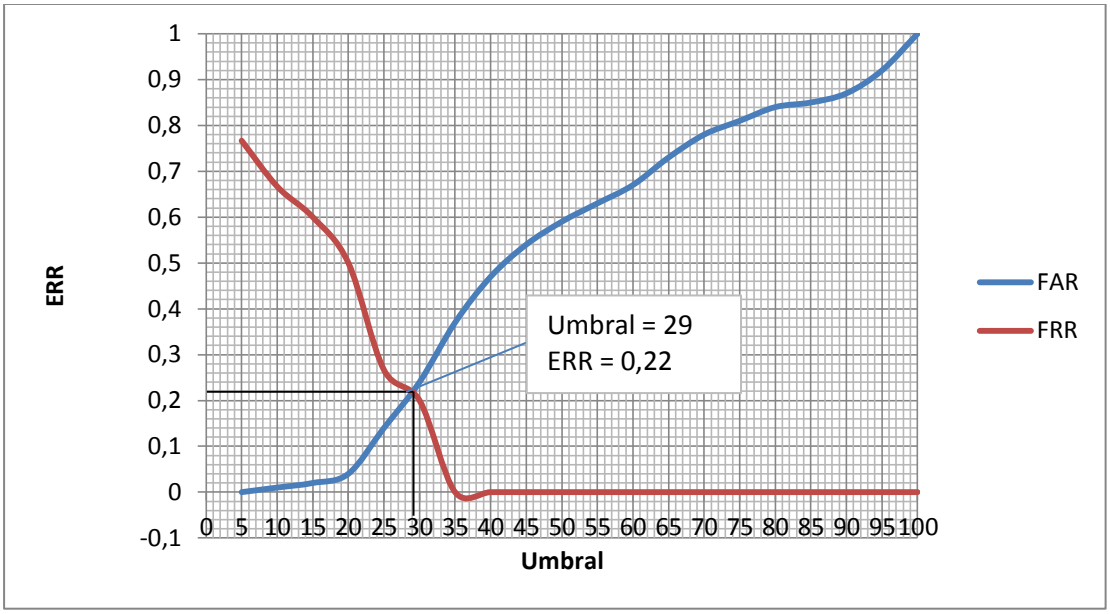

Figura 49. Curvas FAR y FRR con una característica.

Una vez obtenidos todos los resultados con distintos números de características, se observa en la figura 50 la representación de la tasa de error frente al número de características utilizadas para su obtención.

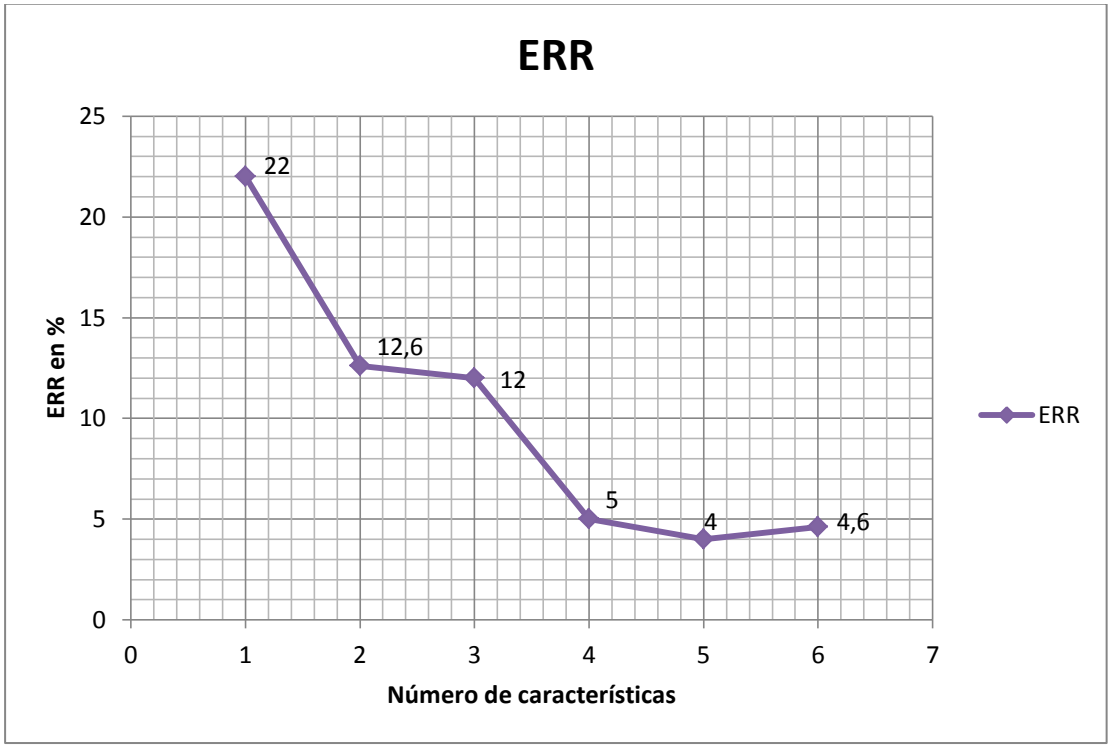

Figura 50. Tasa de error frente a número de características

De la figura anterior se observa como la tasa de error es similar cuando se emplea en el sistema de reconocimiento entre 4 y 6 características, en donde la misma toma valores entre el 4% y el 5%. Mientras que cuando se utilizan entre una y tres características se observa como la tasa de error aumenta de forma considerable llegando alcanzar el 22%, lo que haría que el sistema de reconocimiento fuese poco preciso, aumentando el número de falsos rechazos y falsas aceptaciones. Por lo que existe la posibilidad que el sistema de reconocimiento requiera de un mínimo de cuatro características sin verse afectada la fiabilidad del mismo.

Capítulo 6

Conclusiones

Partiendo de una serie de objetivos se ha desarrollado el sistema de reconocimiento mediante firma manuscrita online 3D, siendo cada uno de los mismos un paso hacia la consecución del proyecto.

Las conclusiones que se han obtenido al finalizar el desarrollo del proyecto, mediante los resultados conseguidos, son las siguientes:

- Se ha demostrado que un dispositivo sencillo, con los sensores adecuados, es capaz de aportar las características que identifiquen a una firma manuscrita online 3D, así como también que las características de las que nos proveía el microcontrolador tenían un poder discriminante alto, como para aumentar la fiabilidad del sistema de reconocimiento.
- Se han diseñado y desarrollado los programas necesarios para el funcionamiento del sistema de reconocimiento. Como el programa de recepción y almacenamiento de datos y el de aplicación del algoritmo DTW, evitando así el uso de programas independientes a nuestro proyecto.
- Teniendo un cierto nivel de conocimiento en los lenguajes de programación, se ha visto cómo es posible la implementación de librerías, previo análisis de las librerías disponibles de código abierto respectivas al algoritmo DTW. Empleando en nuestro programa, la librería que mejor se adapte a las necesidades del mismo, como la disponibilidad de configurar el algoritmo de la distancia empleada para la aplicación del DTW.
- Se ha complementado las funciones de la librería utilizada, como puede ser la normalización de los valores devueltos por cada firma. Por lo que la librería *ndtw* no cuenta con el algoritmo DTW completo, sino que se la complementa con código desarrollado en la aplicación, consiguiendo así el desarrollo total del algoritmo DTW.
- En el análisis de los resultados se ha demostrado como los resultados obtenidos de la tasa de error son bajos, asegurando así unos bajos niveles de las tasas de falsos rechazos y de falsas aceptaciones, aunque se den en un pequeño porcentaje, concretamente menor al 5%.
- Se ha realizado un análisis de la capacidad discriminante de las 6 características que se pueden obtener del acelerómetro (a_x, a_y, a_z) y del giroscopio (ω_x, ω_y, ω_z) de forma gráfica y matemáticamente mediante el índice Fisher.
- En el estudio realizado sobre la posibilidad de que el sistema de reconocimiento utilice un menor número de características se ha demostrado que se pueden utilizar únicamente 4 características sin que esto afecte significativamente a las tasas de error. Esto ayuda a que el sistema propuesto pueda ser implementado en dispositivos de menor capacidad tanto de memoria como de procesamiento.

Capítulo 7

Trabajo futuro

A lo largo del proyecto se ha visto el desarrollo de cada uno de los componentes que forman parte del sistema de reconocimiento de firma manuscrita online 3D y el funcionamiento del mismo, cumpliendo de esta manera los objetivos planteados.

Sin embargo, este proyecto puede seguir evolucionando para ampliar sus funcionalidades o incluso aumentar su nivel de precisión. Por tanto se exponen las ideas con las que se puede seguir trabajando en este proyecto.

7.1 Ampliación de la base de datos

La base de datos utilizada para el desarrollo del proyecto no proviene de ninguna fuente externa, la misma ha sido creada mediante la participación de cinco usuarios. Esto debido a que actualmente no se dispone en Internet de fuentes que pongan a disposición del público en general base de datos para su libre uso, sobre todo respecto a la firma 3D en comparación con la firma 2D, de las que se encuentran varias fuentes y en las que se basan varios de los estudios realizados en las referencias utilizadas.

Por lo que una línea de continuación del proyecto sería la ampliación de la base de datos y obtención de la nueva tasa de error, para de esa manera asegurar una mayor precisión del sistema de reconocimiento, aunque cabe señalar que la tasa de error actual es muy baja, inferior al 5%, pero aun así la misma puede ser reducida o constatada para una base de usuarios mayor.

7.2 Implementación en dispositivos móviles

Se ha visto como mediante el uso de una tarjeta de desarrollo integrada principalmente por un microcontrolador y dos sensores, se ha conseguido obtener las características suficientes para el correcto funcionamiento del sistema de reconocimiento, pero que para completar el mismo se necesitaba de un ordenador, esto debido a la capacidad de procesamiento requerido.

Por lo que otra manera de continuar el proyecto sería la implementación del sistema de reconocimiento en un dispositivo móvil, sin dependencia de una ordenador o equipo similar, para ello existe la necesidad de que el dispositivo cuente con una memoria más amplia y una capacidad de procesamiento mayor, además existe la ventaja de poder utilizar un menor número de características como se ha demostrado durante el desarrollo del proyecto.

7.3 Cifrado de datos

Por último se propone utilizar el cifrado de datos, esto con el objetivo de conseguir una mayor seguridad durante la transmisión de los mismos o incluso frente a programas que intenten acceder a la base de datos, ya que con el cifrado o también conocido como encriptación, se consigue transformar una información legible en información ilegible, disminuyendo así el riesgo de ser leídas por terceras partes. La información ilegible puede ser descifrada mediante una clave del cifrado.

Capítulo 8

Presupuesto

8.1 Coste del personal

En este apartado se detallan la planificación seguida para el desarrollo del proyecto, detalladas en fases e indicando la duración de cada una de ellas.

Fase 0: Estudio de reconocimiento biométrico

- Estudio de reconocimiento biométrico mediante firma manuscrita.
- Estudio del funcionamiento del algoritmo Dynamic Time Warping.

Fase 1: Estudio de la tarjeta de desarrollo STM32F3Discovery

- Recopilación de información relacionada con la tarjeta de desarrollo STM32F3Discovery.
- Estudio del funcionamiento y configuración del microcontrolador STM32F3VCT6 y sensores.
- Aprendizaje de la implementación de los drivers provistos por Atollic TrueStudio.

Fase 2: Desarrollo del Sistema de Reconocimiento

- Desarrollo del programa de recepción y almacenamiento de datos.
- Desarrollo del programa de aplicación del algoritmo DTW y fase de decisión.
- Desarrollo del programa de obtención y envío de datos para el microcontrolador.
- Comprobación del funcionamiento de los programas.

Fase 3: Determinación de la precisión del sistema de reconocimiento

- Construcción de la base de datos.
- Determinación del poder de discriminación de las características obtenidas.
- Análisis de los resultados obtenidos para determinar tasa de error.
- Estudio de la influencia de la variación del número de características sobre la precisión del sistema de reconocimiento.
- Pruebas de funcionamiento del sistema de reconocimiento.

Fase 4: Desarrollo de la memoria

En la tabla 21 se detalla el tiempo dedicado al desarrollo de cada una de las fases.

	Fecha de Inicio	Fecha de Fin	Días dedicados
Fase 0	04/02/2013	14/02/2013	10
Fase 1	15/02/2013	28/02/2013	12
Fase 2	01/03/2013	09/04/2013	28
Fase 3	10/04/2013	07/05/2013	20
Fase 4	08/05/2013	31/05/2013	18
Total			88

Tabla 21. Tiempo de desarrollo por fase

La duración total de dedicación al proyecto es de 88 días, siendo la media de dedicación de 5 horas diarias, se obtiene un total de 440 horas, siendo estas consideradas para la obtención del presupuesto. En la tabla 22 se detalla la parte del presupuesto correspondiente al personal.

	Coste(€/hora)	Duración (horas)	Coste
Fase 0	35	50	1750 €
Fase 1	35	60	2100 €
Fase 2	35	140	4900 €
Fase 3	35	100	3500 €
Fase 4	35	90	3150 €
Total			15400 €

Tabla 22. Coste del personal

8.2 Coste del hardware

En este apartado se detallan los costes producidos por los elementos hardware empleados para el desarrollo del proyecto. Para el cálculo de los mismos se considera una vida útil de 60 meses o 5 años indicado en la tabla como período de depreciación, el tiempo durante los que se utilizó el equipo o dispositivo así como su coste.

El coste imputable de cada hardware se lo calculará con la siguiente fórmula de depreciación:

$$Coste\ de\ depreciación = \frac{A}{B} \cdot C \cdot D \qquad [10]$$

De donde:

- A es el número de meses desde la fecha de facturación en que el equipos es utilizado.
- B el período de depreciación.
- C el coste del equipo sin IVA.
- D el porcentaje de uso que se dedica al proyecto.

Los mismos se encuentran detallados en la tabla 23.

Concepto	Coste	%uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación	Coste Imputable
Ordenador Portátil	750	100	4	60 meses	50 €
Tarjeta de desarrollo STM32F3Discovery	10	100	4	60 meses	0.66 €
Cable USB	5	100	4	60 meses	0.33 €
Total					50.99 €

Tabla 23. Coste del hardware

8.3 Coste del software

En este apartado se detallan los costes asociados a los elementos software utilizados para el desarrollo del proyecto.

El ordenador utilizado cuenta con el sistema operativo Windows 8 Pro de Microsoft, la licencia fue adquirida de forma gratuita a través del programa DreamSpark que pone a disposición

Microsoft mediante un acuerdo con la Universidad Carlos III de Madrid para la comunidad universitaria. La licencia del entorno de desarrollo Visual Studio 2012 también fue adquirida de forma gratuita a través del programa DreamSpark, mientras que la licencia correspondiente a las aplicaciones ofimáticas Microsoft Office 2010, fue adquirida con la reducción de precio correspondiente para estudiantes. En cuanto al entorno de desarrollo Atollic TrueStudio se utilizó la versión Lite, por lo que no ha sido necesario adquirir una licencia de pago. En la tabla 24 se encuentran detallados los costes correspondientes al software.

Concepto	Coste
Licencia Visual Studio 2012	0 €
Licencia Microsoft Office 2010 Profesional	75 €
Entorno de desarrollo Atollic TrueStudio	0 €
Licencia Windows 8 Pro	0 €
Total	75 €

Tabla 24. Costes del software

8.4 Costes varios

Los costes varios son los asociados a gastos tales como la luz, el agua, la calefacción, etc. Por lo que se considera un coste añadido del 10% de la suma de los costes anteriores, obteniendo 1552.60 euros en concepto de costes varios.

8.5 Coste total

El coste total del proyecto incluye todos los apartados anteriores, siendo estos: costes del personal, costes del hardware, costes del software y costes varios. En la tabla 25 se detalla el coste total de estos conceptos y el global del proyecto.

Concepto	Coste Total
Costes del Personal	15400 €
Costes del Hardware	50.99 €
Costes del Software	75 €
Costes Varios	1552.6 €
Total	17078.59€

Tabla 25. Coste total del proyecto

Se concluye que el presupuesto total para la realización del presente proyecto asciende a la cantidad de **17078.59 €**.

Glosario

2D	Two-dimensional
3D	Three-dimensional
a_x	Aceleración lineal en X
a_y	Aceleración lineal en Y
a_z	Aceleración lineal en Z
ADC	Analog to Digital Converter
ALU	Arithmetic Logic Unit
ARM	Advanced RISC Machines
DMA	Direct Memory Access
dps	Degrees per second
DSP	Digital Signal Processor
DTW	Dynamic Time Warping
ERR	Error Rate
FAR	False Accept Rate
FPU	Floating-point Unit
FRR	False Rejection Rate
FR	Fisher Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
MEM	Microelectromechanical Systems
SVM	Support Vector Machines
USB	Universal Serial Bus
ω_x	Velocidad angular en X
ω_y	Velocidad angular en Y
ω_z	Velocidad angular en Z

Referencias

1. Rubesh, P., Bajpai, G., Bhaskar, V.: *3D Signature for Efficient Authentication in Multimodal Biometric Security Systems*. (2010).
2. Sabourin, R., Plamondon, R., Lorette, G.: *Off-line Identification with Handwritten Signature Images: Survey and Perspectives*. In: IAPR workshop on Syntactic and Structural Pattern Recognition, AT&T Murray Hill, New Jersey, pp 377--391, (June 1990).
3. Mendaza, A., Hurtado O., Sánchez, R., Valverde, F., Peláez, C.: *Estudio de reconocimiento biométrico mediante firma manuscrita online basado en SVM usando Análisis Formal de Conceptos*. En: Departamento de Tecnología Eléctronica y Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Leganés, España (2010).
4. Rubesh, P., Bajpai, G., Bhaskar, V.: *Online Multi-Parameter 3D Signature Verification through Curve Fitting*. In: Department and Communication Engineering, SRM University, Kattankulathur, India (May 2009).
5. Alonso, F., Fierrez, J., Martinez, M., Ortega, J.: *Fusion of Static Image and Dynamic Information for Signature Verification*. In: Biometric Recognition Group-ATVS, Escuela Politécnica Superior, Universidad Autonoma de Madrid, Madrid, España (2009).
6. Lejtman, D., George, S.*: On-line Handwritten Signature Verification Using Wavelets and back-propagation Neural Networks*. In: School of Computer and Information Science, University of South Australia (2001).
7. Fernández J. : *Diseño de una Interfaz Gráfica de Evaluación de Algoritmos de Firma Manuscrita*. En: Departamento de Tecnología Electrónica, Universidad Carlos III de Madrid, Leganés, España (Julio 2010).
8. Sánchez, R.*: El Iris Ocular como parámetro para la Identificación Biométrica*. En: Grupo Universitario de Tarjeta Inteligente, Departamento de Tecnología Fotónica, Universidad Politécnica de Madrid, España. (Septiembre 2000).
9. Sanmorino A., Yazid S.: *A Survey for Handwritten Signature Verification*. In: Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia (2012).
10. Plamondon, R., Srihari, S.: *On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey*. In: IEEE Trans. on Pattern Analysisand Machine Intelligence (January 2000).
11. Huang, G., Zhang, D., Zheng, X., Zhu, X.: *An EMG-based Handwriting Recognition through Dynamic Time Warping*. In: State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, China (September 2010).
12. Niels, R.: *Dynamic Time Warping An intuitive way of handwriting recognition?.* In: Faculty of Social Sciences, Department of Artificial Intelligence, Radboud University Nijmegen, The Netherlands. (2004).
13. Lei, H., Palla, S., Govindaraju, V.: *ER^2, An Intuitive Similarity Measure for On-line Signature Verification*. In: Center for Unified Biometrics and Sensors, State University of New York at Buffalo, New York, USA (October 2004).

14. Dong, X., Gu, C., Wang, Z.: *A Local Segmented Dynamic Time Warping Distance Measure Algorithm for Time Series Data Mining*. In: Institute of System Engineering, Tianjin University, China (August 2006).
15. Sun, S., Shi, G.: *Research on the Output Characteristics of MEMS Convective Accelerometer under Heavy Impact*. In: School of Aerospace Scienci and Engineering, Beijing Institute of Technology, Beijing, China (January 2009).
16. Ji, X., Wang, S., Xu, Y., Xia, D.: *Application of the Digital Signal Procession in the MEMS Gyroscope De-drift*. In: Department of Instruments Science and Engineering, Southeast University, China (January 2006).
17. Hurtado O. M.: *Online Signature Verification Algorithms and Development of Signature International Standards*. In: Departamento de Tecnología Electrónica, Universidad Carlos III de Madrid, Leganés, España (Septiembre 2011).
18. Jain A. K., Duin R., Mao J.: *Statiscal Pattern Recognition: A Review*. In: Department of Computer Science and Engineering, Michigan State University, East Lansing, USA and Pattern Recognition Group, Department of Applied Physics, Delft University of Technology, The Netherlands (November 1999).
19. Pascual Gaspar J.M.: *Uso de la Firma Manuscrita Dinámica para el Reconocimiento Biométrico de Personas en Escenarios Prácticos*. En: Departamento de Informática, Escuela Técnica Superior de Ingeniería Informática, Universidad de Valladolid.
20. Zhang D.: *Automated Biometrics: Technologies and Systems*. In: Kluwer Academic Publishers (2000).